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 №1.4**

по дисциплине: Дискретная математика

тема: «Теоретико-множественные уравнения»

Выполнил: ст. группы ПВ-211

Чувилко Илья Романович

Проверили:

Рязанов Юрий Дмитриевич

Белгород 2022 г.

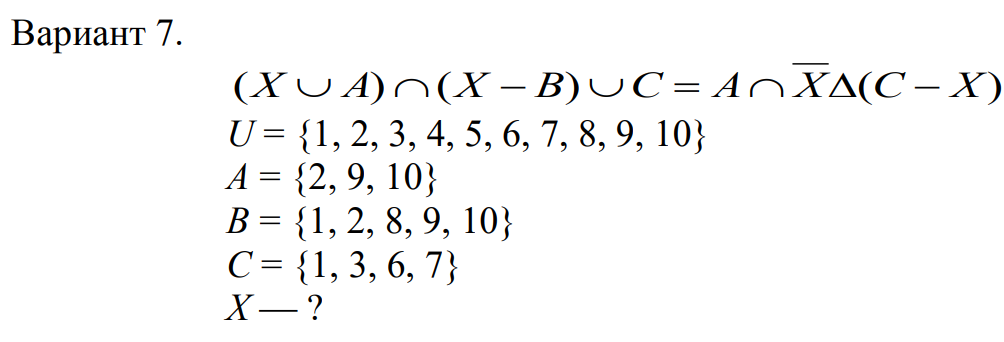
**Цель занятия:** научиться решать теоретико-множественные уравне-

ния с применением ЭВМ.

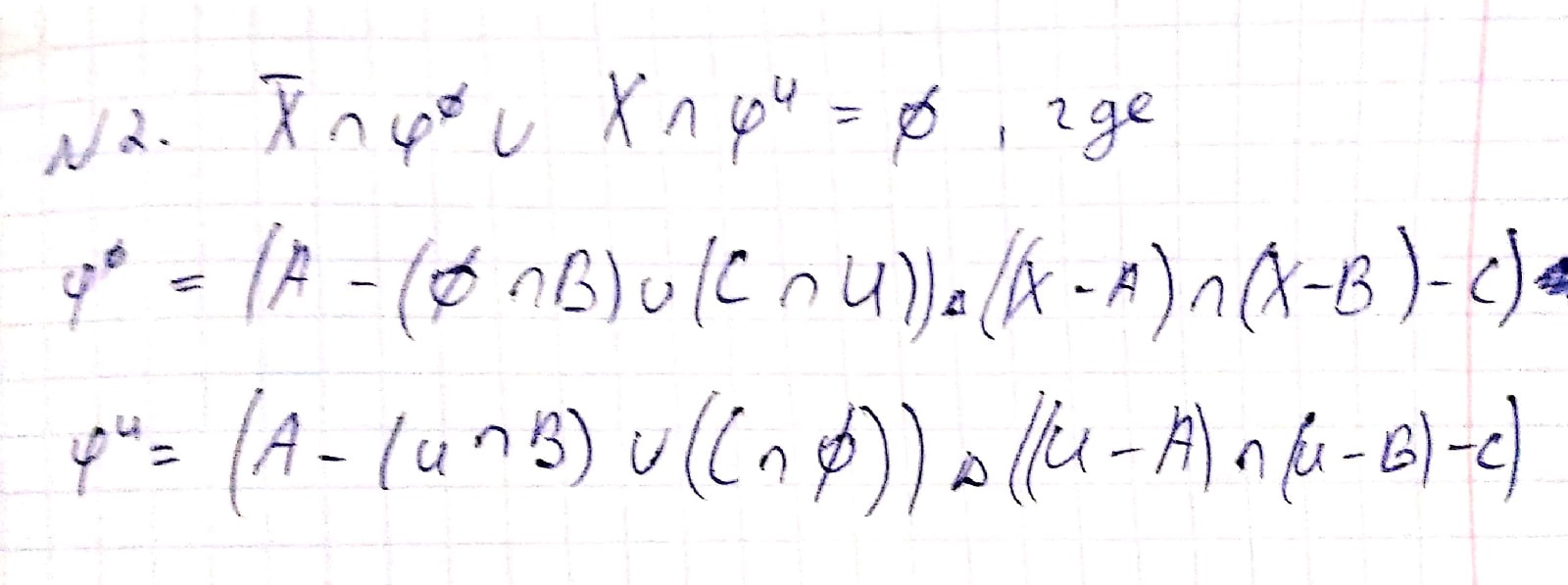
Вариант 7

**Задания**

1. Преобразовать исходное уравнение в уравнение с пустой правой частью.



2. Преобразовать левую часть уравнения к виду X    X  U , используя разложение Шеннона по неизвестному множеству X.



3. Написать программу, вычисляющую значения множеств  и U при заданных исходных множествах.

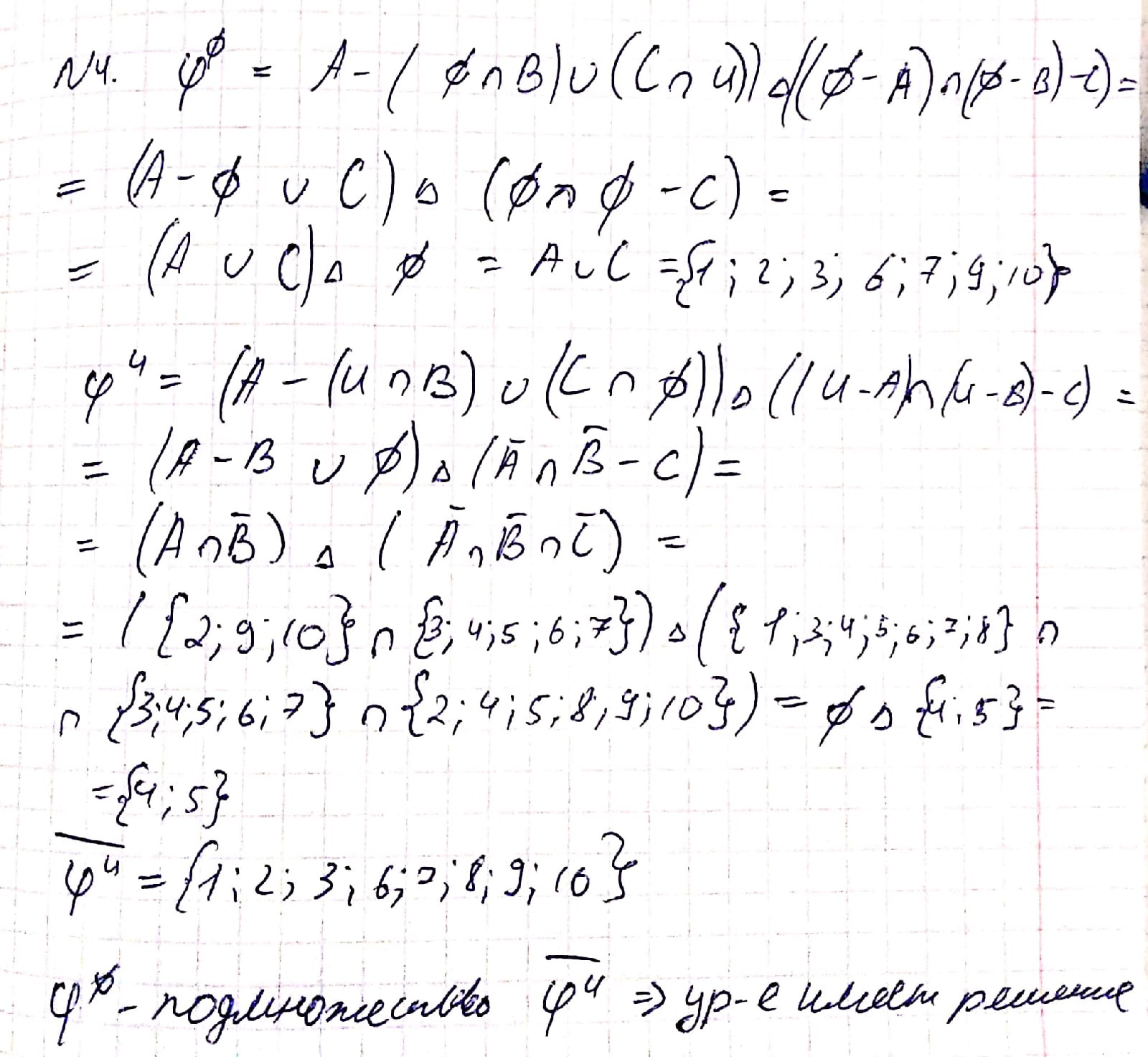
class InToPost { private: stack<char> stackOperators; string postfixNotation; string input; public: explicit InToPost(string input) : input(std::move(input)) {}; void gotOper(char oper, int priorityNew) { while (!this->stackOperators.empty()) { char topOperator = this->stackOperators.top(); if (topOperator != '(') { int priorityTop; if (topOperator == '!') priorityTop = 1; else if (topOperator == '&') priorityTop = 2; else priorityTop = 3; if (priorityNew >= priorityTop) { this->postfixNotation += topOperator; this->stackOperators.pop(); } else break; } else { break; } } this->stackOperators.push(oper); } void gotParen() { while (!this->stackOperators.empty()) { char i = this->stackOperators.top(); this->stackOperators.pop(); if (i == u'(') break; else this->postfixNotation += i; } } void makePostfixNotation(const string &infixNotation) { for (auto i: infixNotation) { switch (i) { case 'u': case '^': case '-':{ gotOper(i, 3); break; } case '&':{ gotOper(i, 2); break; } case '!': { gotOper(i, 1); break; }

case '(': { this->stackOperators.push(i); break; } case ')': { gotParen(); break; } default: postfixNotation += i; } } while (!this->stackOperators.empty()) { this->postfixNotation += this->stackOperators.top(); this->stackOperators.pop(); } } string getPostfixNotation() { makePostfixNotation(input); return this->postfixNotation; } }; class SolverPost { private: int countSets; string postExpression; unordered\_array\_set X; int a[3] = {2,9,10}; int b[5] = {1,2,8,9,10}; int c[4] = {1,3,6,7}; int d[10] = {1,2,3,4,5,6,7,8,9,10}; unordered\_array\_set universum = UOAS\_create\_from\_array(d, 10); vector<unordered\_array\_set> sets = { UOAS\_create\_from\_array(a, 3), UOAS\_create\_from\_array(b, 5), UOAS\_create\_from\_array(c, 4), }; stack<unordered\_array\_set> executionStack; public: void inputX(unordered\_array\_set x) { this->X = x; } unordered\_array\_set getUniversum() { return universum; } SolverPost(int numSets, string postExpression) : countSets(numSets), postExpression(std::move(postExpression)) {}; unordered\_array\_set fi() { while (!this->executionStack.empty()) this->executionStack.pop(); int curIndex = 0; while (curIndex < postExpression.size()) {

char curSymbol = this->postExpression[curIndex]; if ('A' <= curSymbol && curSymbol <= 'Z') { if (curSymbol == 'U') this->executionStack.push(this->universum); else if (curSymbol == 'X') this->executionStack.push(this->X); else this->executionStack.push(this->sets[curSymbol - 'A']); } else { if (curSymbol == 'u') { unordered\_array\_set set1 = this->executionStack.top(); this->executionStack.pop(); unordered\_array\_set set2 = this->executionStack.top(); this->executionStack.pop(); this->executionStack.push(UOAS\_union(set1, set2)); } else if (curSymbol == '&') { unordered\_array\_set set1 = this->executionStack.top(); this->executionStack.pop(); unordered\_array\_set set2 = this->executionStack.top(); this->executionStack.pop(); this->executionStack.push( UOAS\_intersection(set1, set2)); } else if (curSymbol == '^') { unordered\_array\_set set1 = this->executionStack.top(); this->executionStack.pop(); unordered\_array\_set set2 = this->executionStack.top(); this->executionStack.pop(); this->executionStack.push( UOAS\_symmetricDifference(set1, set2)); } else if (curSymbol == '-') { unordered\_array\_set set1 = this->executionStack.top(); this->executionStack.pop(); unordered\_array\_set set2 = this->executionStack.top(); this->executionStack.pop(); this->executionStack.push(UOAS\_difference(set2, set1)); } else if (curSymbol == '!') { unordered\_array\_set set1 = this->executionStack.top(); this->executionStack.pop(); this->executionStack.push( UOAS\_complement(set1, this->universum)); } } curIndex++; } return executionStack.top(); } };

int main() {#ifdef FIRST string input = "(A-(X&B)u(C&!X))^((X-A)&(X-B)-C)"; InToPost translator(input); SolverPost solver(3, translator.getPostfixNotation()); int a[0]; cout << translator.getPostfixNotation() << '\n'; solver.inputX(UOAS\_create\_from\_array(a, 0)); unordered\_array\_set set = solver.fi(); qsort(set.data, set.size, sizeof(int), compare\_ints); for (int i = 0; i < set.size; i++) { cout << set.data[i] << " "; } cout << endl; InToPost translator2(input); SolverPost solver2(3, translator2.getPostfixNotation()); int b[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}; solver2.inputX(UOAS\_create\_from\_array(b, 10)); unordered\_array\_set set2 = solver2.fi(); set2 = UOAS\_complement(set2, UOAS\_create\_from\_array(b, 10)); qsort(set2.data, set2.size, sizeof(int), compare\_ints); for (int i = 0; i < set2.size; i++) { cout << set2.data[i] << " "; }#endif return 0; }

4. Вычислить значения множеств  и U и сделать вывод о существовании решения уравнения. Если решения уравнения не существует, то выполнить п.п. 1—4 для следующего (предыдущего) варианта.



5. Определить мощность общего решения, найти некоторые (или все) частные решения, в том числе частные решения наименьшей и наибольшей мощности.

**Общее решение:** {{},{1},{2},{1,2},{3},{1,3},{2,3},{1,2,3},{6},{1,6},{2,6},{1,2,6},{3,6},{1,3,6},{2,3,6},{1,2,3,6},{7},{1,7},{2,7},{1,2,7},{3,7},{1,3,7},{2,3,7},{1,2,3,7},{6,7},{1,6,7},{2,6,7},{1,2,6,7},{3,6,7},{1,3,6,7},{2,3,6,7},{1,2,3,6,7},{8},{1,8},{2,8},{1,2,8},{3,8},{1,3,8},{2,3,8},{1,2,3,8},{6,8},{1,6,8},{2,6,8},{1,2,6,8},{3,6,8},{1,3,6,8},{2,3,6,8},{1,2,3,6,8},{7,8},{1

,7,8},{2,7,8},{1,2,7,8},{3,7,8},{1,3,7,8},{2,3,7,8},{1,2,3,7,8},{6,7,8},{1,6,7,8},{2,6,7,8},{1,2,6,7,8},{3,6,7,8},{1,3,6,7,8},{2,3,6,7,8},{1,2,3,6,7,8},{9},{1,9},{2,9},{1,2,9},{3,9},{1,3,9},{2,3,9},{1,2,3,9},{6,9},{1,6,9},{2,6,9},{1,2,6,9},{3,6,9},{1,3,6,9},{2,3,6,9},{1,2,3,6,9},{7,9},{1,7,9},{2,7,9},{1,2,7,9},{3,7,9},{1,3,7,9},{2,3,7,9},{1,2,3,7,9},{6,7,9},{1,6,7,9},{2,6,7,9},{1,2,6,7,9},{3,6,7,9},{1,3,6,7,9},{2,3,6,7,9},{1,2,3,6,7,9},{8,9},{1,8,9},{2,8,9},{1,2,8,9},{3,8,9},

{1,3,8,9},{2,3,8,9},{1,2,3,8,9},{6,8,9},{1,6,8,9},{2,6,8,9},{1,2,6,8,9},{3,6,8,9},{1,3,6,8,9},{2,3,6,8,9},{1,2,3,6,8,9},{7,8,9},{1,7,8,9},{2,7,8,9},{1,2,7,8,9},{3,7,8,9},{1,3,7,8,9},{2,3,7,8,9},{1,2,3,7,8,9},{6,7,8,9},{1,6,7,8,9},{2,6,7,8,9},{1,2,6,7,8,9},{3,6,7,8,9},{1,3,6,7,8,9},{2,3,6,7,8,9},{1,2,3,6,7,8,9},{10},{1,10},{2,10},{1,2,10},{3,10},{1,3,10},{2,3,10},{1,2,3,10},{6,10},{1,6,10},{2,6,10},{1,2,6,10},{3,6,10},{1,3,6,10},{2,3,6,10},{1,2,3,6,10},{7,10},{1,7,10},{2,7,10,

},{1,2,7,10},{3,7,10},{1,3,7,10},{2,3,7,10},{1,2,3,7,10},{6,7,10},{1,6,7,10},{2,6,7,10},{1,2,6,7,10},{3,6,7,10},{1,3,6,7,10},{2,3,6,7,10},{1,2,3,6,7,10},{8,10},{1,8,10},{2,8,10},{1,2,8,10},{3,8,10},{1,3,8,10},{2,3,8,10},{1,2,3,8,10},{6,8,10},{1,6,8,10},{2,6,8,10},{1,2,6,8,10},{3,6,8,10},{1,3,6,8,10},{2,3,6,8,10},{1,2,3,6,8,10},{7,8,10},{1,7,8,10},{2,7,8,10},

{1,2,7,8,10},{3,7,8,10},{1,3,7,8,10},{2,3,7,8,10},{1,2,3,7,8,10},{6,7,8,10},{1,6,7,8,10},{2,6,7,8,10},{1,2,6,7,8,10},{3,6,7,8,10},{1,3,6,7,8,10},{2,3,6,7,8,10},{1,2,3,6,7,8,10},{9,10},{1,9,10},{2,9,10},{1,2,9,10},{3,9,10},{1,3,9,10},{2,3,9,10},{1,2,3,9,10},{6,9,10},{1,6,9,10},{2,6,9,10},{1,2,6,9,10},{3,6,9,10},{1,3,6,9,10},{2,3,6,9,10},{1,2,3,6,9,10},{7,9,10

},{1,7,9,10},{2,7,9,10},{1,2,7,9,10},{3,7,9,10},{1,3,7,9,10},{2,3,7,9,10},{1,2,3,7,9,10},{6,7,9,10},{1,6,7,9,10},{2,6,7,9,10},{1,2,6,7,9,10},{3,6,7,9,10},{1,3,6,7,9,10},{2,3,6,7,9,10},{1,2,3,6,7,9,10},{8,9,10},{1,8,9,10},{2,8,9,10},{1,2,8,9,10},{3,8,9,10},{1,3,8,9,10},{2,3,8,9,10},{1,2,3,8,9,10},{6,8,9,10},{1,6,8,9,10},{2,6,8,9,10},{1,2,6,8,9,10},{3,6,8,9,10},{1,3,6,8,9,10},{2,3,6,8,9,10},{1,2,3,6,8,9,10},{7,8,9,10},{1,7,8,9,10},{2,7,8,9,10},{1,2,7,8,9,10},{3,7,8,9,10},{1,3,7,8,9,10},{2,3,7,8,9,10},{1,2,3,7,8,9,10},{6,7,8,9,10},{1,6,7,8,9,10},{2,6,7,8,9,10},{1,2,6,7,8,9,10},{3,6,7,8,9,10},{1,3,6,7,8,9,10},{2,3,6,7,8,9,10},{1,2,3,6,7,8,9,10}

**Мощность общего решения:** 256.

**Минимальное по мощности частное решение:** .

**Максимальное по мощности частное решение:** {1,2,3,6,7,8,9,10}.

6. Написать программу для проверки найденных решений.

int main() {#ifdef SECOND int a[] = {1,2,3,6,7,8,9,10}; unordered\_array\_set x = UOAS\_create\_from\_array(a, 8); string input = "(A-(X&B)u(C&!X))^((X-A)&(X-B)-C)"; InToPost translator(input); SolverPost solver(8, translator.getPostfixNotation()); solver.inputX(x); unordered\_array\_set set = solver.fi(); if (set.size == 0) cout << "Is this solution"; else cout << "Is not a solution"; cout << endl;#endif return 0;}