Susan Emberton - sz012

Julian Shen – sz031

Group 4

Discrete Structures

Project 1

Source Code:

#include <iostream>

using namespace std;

void problem1();

void problem2();

void problem3();

void problem4();

void process1(int, int);

void process2(int, int);

void process3(int, int);

void process4(int, int, int);

int biDir(int, int);

int disj(int, int);

int nDisjN(int, int);

int ifThen(int, int);

int conj(int, int);

int nConj(int, int);

int taut = 1;

int main()

{

    problem1();

    problem2();

    problem3();

    problem4();

}

void problem1()

{

    int p = 1;

    int q = 1;

    cout << endl

         << "Check equivalence: p <-> q === ((p + q) \* (~p + ~q))" << endl;

    cout << "  p | q | p + q | ~p + ~q | p <-> q | <-> | (p + q) \* (~p + ~q)" << endl;

    process1(p, q);

    q = 0;

    process1(p, q);

    p = 0;

    q = 1;

    process1(p, q);

    q = 0;

    process1(p, q);

    if (taut)

    {

        cout << "p <-> q is equivalent to ((p + q) \* (~p + ~q))" << endl;

    }

    else

    {

        cout << "p <-> q is not equivalent to ((p + q) \* (~p + ~q))" << endl;

    }

    taut = 1;

}

void process1(int p, int q)

{

    int a, b, c, d, equality;

    a = disj(p, q);

    b = nDisjN(p, q);

    c = biDir(p, q);

    d = conj(a, b);

    equality = biDir(c, d);

    if (!equality)

        taut = 0;

    cout << "  " << p << "   " << q << "     " << a << "        " << b << "         " << c;

    cout << "       " << equality << "            " << d << endl;

}

void problem2()

{

    int p = 1;

    int q = 1;

    cout << endl

         << "Check equivalence: ~(p \* q) === ~p + ~q" << endl;

    cout << "  p | q | ~(p \* q) | <-> | ~p + ~q" << endl;

    process2(p, q);

    q = 0;

    process2(p, q);

    p = 0;

    q = 1;

    process2(p, q);

    q = 0;

    process2(p, q);

    if (taut)

    {

        cout << "~(p \* q) is equivalent to ~p + ~q" << endl;

    }

    else

    {

        cout << "~(p \* q) is not equivalent to ~p + ~q" << endl;

    }

    taut = 1;

}

void process2(int p, int q)

{

    int a, b, equality;

    a = nConj(p, q);

    b = nDisjN(p, q);

    equality = biDir(a, b);

    if (!equality)

        taut = 0;

    cout << "  " << p << "   " << q << "       " << a << "       " << equality << "       " << b << endl;

}

void problem3()

{

    int p = 1;

    int q = 1;

    cout << endl

         << "Check equivalence: p <-> q === ((p -> q) \* (q -> p))" << endl;

    cout << "  p | q | p -> q | q -> p | p <-> q | <-> | (p -> q) \* (q -> p)" << endl;

    process3(p, q);

    q = 0;

    process3(p, q);

    p = 0;

    q = 1;

    process3(p, q);

    q = 0;

    process3(p, q);

    if (taut)

    {

        cout << "p <-> q is equivalent to ((p -> q) \* (q -> p))" << endl;

    }

    else

    {

        cout << "p <-> q is not equivalent to ((p -> q) \* (q -> p))" << endl;

    }

    taut = 1;

}

void process3(int p, int q)

{

    int a, b, c, d, equality;

    a = ifThen(p, q);

    b = ifThen(q, p);

    c = biDir(p, q);

    d = conj(a, b);

    equality = biDir(c, d);

    if (!equality)

        taut = 0;

    cout << "  " << p << "   " << q << "      " << a << "        " << b << "        " << c;

    cout << "       " << equality << "             " << d << endl;

}

void problem4()

{

    int p = 1;

    int q = 1;

    int r = 1;

    cout << endl

         << "Check equivalence: (p -> q) -> r === p -> (q -> r)" << endl;

    cout << "  p | q | r | p -> q | q -> r | (p -> q) -> r | <-> | p -> (q -> r)" << endl;

    process4(p, q, r);

    r = 0;

    process4(p, q, r);

    r = 1;

    q = 0;

    process4(p, q, r);

    r = 0;

    process4(p, q, r);

    p = 0;

    q = 1;

    r = 1;

    process4(p, q, r);

    r = 0;

    process4(p, q, r);

    p = 0;

    q = 0;

    r = 1;

    process4(p, q, r);

    r = 0;

    process4(p, q, r);

    if (taut)

    {

        cout << "(p -> q) -> r is equivalent to p -> (q -> r)" << endl;

    }

    else

    {

        cout << "(p -> q) -> r is not equivalent to p -> (q -> r)" << endl;

    }

    taut = 1;

}

void process4(int p, int q, int r)

{

    int a, b, c, d, equality;

    a = ifThen(p, q);

    b = ifThen(q, r);

    c = ifThen(a, r);

    d = ifThen(p, b);

    equality = biDir(c, d);

    if (!equality)

        taut = 0;

    cout << "  " << p << "   " << q << "   " << r << "     " << a << "        " << b << "             " << c;

    cout << "         " << equality << "         " << d << endl;

}

// subroutines

int biDir(int p, int q)

{

    if (p == q)

        return 1;

    else

        return 0;

}

int disj(int p, int q)

{

    if (p || q)

        return 1;

    else

        return 0;

}

int nDisjN(int p, int q)

{

    if (p == 0)

        p = 1;

    else if (p == 1)

        p = 0;

    if (q == 0)

        q = 1;

    else if (q == 1)

        q = 0;

    return disj(p, q);

}

int ifThen(int p, int q)

{

    if (p && !q)

        return 0;

    else

        return 1;

}

int conj(int p, int q)

{

    if (p && q)

        return 1;

    else

        return 0;

}

int nConj(int p, int q)

{

    if (p && q)

        return 0;

    else

        return 1;

}

Output:

