МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра информатики и систем управления

Лабораторная работа №1

по дисциплине Интеллектуальные системы

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Дмитриев Д.В.

(подпись)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Смирнов А.В.

(подпись)

Группа М17-ИВТ-3

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород

2017

**Задание № 14**

Даны два натуральных числа, больших единицы, но меньших ста. Сумма этих чисел дана господину *S*, а произведение дано господину *Р*. Господин *Р* звонит господину *S* и говорит:

*Р*: Я не могу найти эти два числа .

*S*: Я знаю, что у вас и не хватило бы ума.

*Р*: Ах, так... Но тогда я их знаю!

*S*: Ну, тогда я тоже и знаю!

Необходимо по данному диалогу найти эти числа.

**Решение:**

Перенумеруем высказывания господ:

*P*: "Я не могу определить эти числа" (1)

*S*: "Я знаю, что Вы не можете этого сделать" (2)

*P*: "Но тогда я знаю эти числа" (3)

*S*: "Тогда и я знаю эти числа" (4)

Задуманные натуральные числа n и m удовлетворяют условиям

2 <= n <= m <= 99 (5)

Такие числа для краткости будем называть допустимыми.

Пусть X := n\*m и Y := n+m.

Введем две специальные функции, значениями которых будут множества чисел:

p(x) - множество произведений пар допустимых чисел, сумма которых равна x.

Например,

p(8) = { 12, 15 },

p(10) = { 16, 21, 24, 25 }.

s(x) - множество сумм пар допустимых чисел, произведение которых равно x.

Например,

s(11) - пустое множество,

s(15) = { 8 },

s(24) = { 14, 11, 10 }.

Утверждение (1) в переводе на наши обозначения означает следующее:

s(X) содержит более одного элемента. (1')

Обозначим через P1 множество чисел, удовлетворяющих (1').

Начало этого множества несложно выписать явно:

P1 = { 12 16 18 20 24 28 30 32 36 40 42 45 48 ... }

Как легко видеть, в P1 нет чисел, являющихся произведением двух простых чисел (а также кубом простого числа).

Отметим еще такой факт, который нам дальше понадобится: P1 не содержит чисел, кратных 53. Действительно для допустимого числа t s(53t) = { 53+t }, то есть состоит из одного числа так как 106 - следующее кратное числу 53 - уже не является допустимым.

Утверждение (2) также легко перевести:

p(Y) является подмножеством P1. (2')

Обозначим множество чисел, удовлетворяющих (2'), через S1. В частности, отсюда следует, что S1 не содержит чисел, представимых в виде суммы двух простых.

Известная гипотеза Гольдбаха-Эйлера (не доказанная, но проверенная для всех досягаемых на ЭВМ чисел) утверждает, что каждое четное число, начиная с 4, является суммой двух простых. Следовательно, S1 не содержит четных чисел. Кроме того, S1 не содержит чисел вида p+2, где p - простое.

Таким образом, в качестве кандидатов в элементы S1 остаются только такие числа:

{ 11 17 23 27 29 35 37 41 47 51 53 57 59 ... }

Выбросим отсюда все числа, большие 55: Если Y>55, то Y можно представить в виде суммы 53 + (Y-53), поэтому число 53\*(Y-53) принадлежит s(Y), но произведение таких чисел, как отмечено выше, не принадлежит P1.

Теперь проверке в S1 подлежат только числа:

{ 11 17 23 27 29 35 37 41 47 51 53 }.

После фразы (2) P уже знает, что число Y нечетное. Поэтому он может рассматривать в качестве пары сомножителей уже не любые делители числа X, а только делители разной четности. В частности, если X=2^n\*p, где p - простое, то s(X) содержит единственное нечетное число Y=2^n+p, и P сможет однозначно определить загаданные числа - это 2^n и p.

Фраза (3) означает, что

пересечение множества s(X) и множества S1 состоит ровно из одного элемента. (3')

Обозначим через P2 множество тех X из P1, для которых выполнено свойство (3'). В частности, 16 не принадлежит P2, так как s(16) состоит только из четных чисел. Так как 30 = 5\*6 = 2\*15 и оба числа 5+6 и 2+15 принадлежат S1, то пересечение s(30) и S1 содержит хотя бы два элемента, т.е. 30 тоже не принадлежит P2. Рассуждая аналогично, можно получить первые элементы P2:

P2 = { 18 24 28 52 72 76 96 100 102 110 ... }

Что можно почерпнуть из фразы (4), сказанной господином S ? Что

пересечение p(Y) и множества P2 состоит ровно из одного элемента X (4')

(иначе S не смог бы выбрать между двумя или более вариантами).

Пусть S2 - множество тех Y из S1, для которых выполнено условие (4').

В отыскании множества S2 и состоит, собственно, задача (для каждого Y из S2 мы однозначно определяем X из условия (4'), а зная X и Y, находим загаданные числа).

=>

**Программа на языке Prolog (интерпретатор SWI-Prolog)**

% equation - поиск корней квадратного уравнения

delta(A, B, C, D):- D is B\*B - 4\*A\*C.

equation(A,B,C,X) :-

delta(A,B,C,D1),

( D1 < 0

-> X is 0

; D1 =:= 0

-> X is -B/2\*A

; X1 is (-B+sqrt(D1))/2\*A

, X2 is (-B-sqrt(D1))/2\*A

, X = [X1,X2]

).

/\*

найти множество L, элементом которого является T,

такое, что

среди чисел от 2 до Х

есть числа А и В, входящие в этот диапазон

такие что А не равно В,

сумма А и В равна Х

произведедение А и В и есть Т.

A^B^Goal нужно чтобы все результаты сложились в один лист

\*/

p(X, L):-

setof(T,

A^B^(numlist(2,X,NL),

member(A,NL),member(B,NL),

A=\=B, X is A + B, T is A\*B),

L), !; L = [].

/\*

найти множество L, элементом которого является T,

такое, что

среди чисел от 2 до Х

есть числа А и В, входящие в этот диапазон

такие что А не равно В,

произведение А и В равна Х

сумма А и В и есть Т.

При этом, если L - пустое множество, вернуть пустой список.

Сделано так:

сразу после setof идет оператор "," (И) и дальше оператор "!" (Cut),

Оператор ! работает так:

если стоящее слева от него выражение истинно, дальше проверка не идет

если бы его там не было, а просто стояла точка с запятой, тогда всегда было бы два решения:

множество L и пустой список, либо если L - пустое множество, тогда только пустой список

с помощью ! мы всегда будем получать только одно решение: либо множество L, либо [].

\*/

s(X, L):-

setof(T,

A^B^(numlist(2,X,NL),

member(A,NL),member(B,NL),

A=\=B, X is A \* B, T is A+B),

L), !;

L= [].

/\*

(1) P: "Я не могу определить эти числа"

Утверждение (1) в переводе на наши обозначения означает следующее:

s(X) содержит более одного элемента. (1')

Обозначим через P1 множество чисел, удовлетворяющих (1').

P1 вычисляется так:

P1 есть множество элементов L, таких, что

множество s(L) состоит более чем из одного элемента

\*/

getP1(L):-

findall(Y,

(numlist(2, 800, NL),

member(Y, NL),

s(Y, L),

length(L, Len),

Len > 1),

L).

/\*

(2) S: "Я знаю, что Вы не можете этого сделать"

означает, что

p(Y) является подмножеством P1.

Все подходящие под это утверждение Y занесем в S1.

\*/

getS1(L):-

getP1(P1),

findall(Y,

(numlist(2,800,NL),

member(Y, NL),

p(Y, L1),

subset(L1,P1)),

L).

/\*

(3) P: "Но тогда я знаю эти числа"

В P2 попадают только такие Х, для которых

пересечение множества s(X) и множества S1 состоит ровно из одного элемента

это означает, что если есть только один элемент в множестве s(X)

который встречается в S1, тогда Х попадает в Р2.

\*/

getP2(L):-

getS1(S1),

findall(Y,

(numlist(2, 800, NL),

member(Y, NL),

s(Y, Si),

intersection(Si, S1, Intersec),

length(Intersec, Len),

Len = 1),

L).

/\*

(4) S: "Тогда и я знаю эти числа"

означает, что

пересечение p(Y) и множества P2 состоит ровно из одного элемента X

(иначе господин S не смог бы выбрать между двумя или более вариантами).

Пусть S2 - множество тех Y из S1, для которых выполнено условие (4').

В отыскании множества S2 и состоит, собственно, задача (для каждого Y из S2

мы однозначно определяем X из условия (4'), а зная X и Y,

находим загаданные числа, решая квадратное уравнение).

\*/

getS2(L):-

getS1(S1),

getP2(P2),

findall(K,

(member(Y, S1),

p(Y, Pi),

intersection(Pi, P2, Intersec),

length(Intersec, Len),

Len = 1,

T is Intersec,

K = [Y, T]),

[L|\_]).

solve(A):-

getS2([H|T]),

equation(1,-H, T, A).

**Результаты**

