CORRESPONDENCE BETWEEN THE SOLUTIONS OF AN EQUATION AND THE DIVISORS OF NATURAL NUMBERS

Edoardo GUEGLIO

egueglio@gmail.com

There is a correspondence between the positive solutions of a diophantine equation and the divisors of natural numbers

Premise

All this research has been done using Mathematica©, a symbolic computation environment that uses a programming language called Wolfram Language.

I investigated the positive solutions of Diophantine equation x+y+x*y+x^2=n where n is a natural number for convenience starting from 2. In terms of Mathematica all this can be expressed as:

```
Reduce[x+y+x*y+y^2==n&&x>0&&y>0,{x,y},Integers]
```

Reduce is the function used to solve diophantine equations and has three arguments:

- the set of conditions that must be satisfied by the variables joined together by the symbol && which stands for 'and'
- the set of variable names (in this case x and y)
- the domain of the values (in this case the domain of integers)

To complete the informations needed to understand the following we have another functions:

```
Divisors[n]
```

This function given a natural n returns the list expressed as a comma separated sequence of values between braces that starts from 1 and goes to n. Example:

```
In[ ]:= Divisors[30]
Out[ ]= {1, 2, 3, 5, 6, 10, 15, 30}
```

Now lets try to calculate the solutions of the above equation replacing n with 30:

```
In[ ]:= Reduce[x+y+x*y+y^2==30&&x>0&&y>0,{x,y},Integers]
        Divisors[30]

Out[ ]= (x == 2 && y == 4) || (x == 8 && y == 2) || (x == 14 && y == 1)

Out[ ]= {1, 2, 3, 5, 6, 10, 15, 30}
```

What we get is a list of solutions, each enclosed in round brackets and separated by symbol || which stands for ‘or’ followed by the list of divisors of 30. **The interesting thing is that if you calculate in each solution x+y you get the same values in the second half of divisors list except the last number.**

Following this concept what is happen when we use for n a prime number? A prime number p has the divisors list like {1,p} and hence the second half except last corresponds to empty solutions list:

```
In[ ]:= Reduce[x+y+x*y+y^2==31&&x>0&&y>0,{x,y},Integers]
        Divisors[31]

Out[ ]= False

Out[ ]= {1, 31}
```

Output False means that there are no solutions.

We have used above number 30 which has an even number of divisors. But what happens with a number with odd number of divisors:

```
In[ ]:= Reduce[x+y+x*y+y^2==36&&x>0&&y>0,{x,y},Integers]
        Divisors[36]

Out[ ]= (x == 1 && y == 5) || (x == 6 && y == 3) || (x == 10 && y == 2) || (x == 17 && y == 1)

Out[ ]= {1, 2, 3, 4, 6, 9, 12, 18, 36}
```

We don’t start from the second half but from the number in the middle of the sequence.

Now we start with n=4 to check what we have verified. Obviously this is true for n<4 because we have four terms in the expression (x+y+x*y+y^2) and x>0 and y>0 so each term will be at last 1.

```
In[ ]:= Reduce[x+y+x*y+y^2==4&&x>0&&y>0,{x,y},Integers]
        Divisors[4]

Out[ ]= x == 1 && y == 1

Out[ ]= {1, 2, 4}
```

```
In[ ]:= Reduce[x+y+x*y+y^2==5&&x>0&&y>0,{x,y},Integers]
        Divisors[5]

Out[ ]= False

Out[ ]= {1, 5}
```

In[ ]:=
```
Reduce[x+y+x*y+y^2==6&&x>0&&y>0,{x,y},Integers]
Divisors[6]
```
Out[ ]= x == 2 && y == 1

Out[ ]= {1, 2, 3, 6}

In[ ]:=
```
Reduce[x+y+x*y+y^2==7&&x>0&&y>0,{x,y},Integers]
Divisors[7]
```
Out[ ]= False

Out[ ]= {1, 7}

In[ ]:=
```
Reduce[x+y+x*y+y^2==8&&x>0&&y>0,{x,y},Integers]
Divisors[8]
```
Out[ ]= x == 3 && y == 1

Out[ ]= {1, 2, 4, 8}

In[ ]:=
```
Reduce[x+y+x*y+y^2==9&&x>0&&y>0,{x,y},Integers]
Divisors[9]
```
Out[ ]= x == 1 && y == 2

Out[ ]= {1, 3, 9}

In[ ]:=
```
Reduce[x+y+x*y+y^2==10&&x>0&&y>0,{x,y},Integers]
Divisors[10]
```
Out[ ]= x == 4 && y == 1

Out[ ]= {1, 2, 5, 10}

In[ ]:=
```
Reduce[x+y+x*y+y^2==11&&x>0&&y>0,{x,y},Integers]
Divisors[11]
```
Out[ ]= False

Out[ ]= {1, 11}

In[ ]:=
```
Reduce[x+y+x*y+y^2==12&&x>0&&y>0,{x,y},Integers]
Divisors[12]
```
Out[ ]= (x == 2 && y == 2) || (x == 5 && y == 1)

Out[ ]= {1, 2, 3, 4, 6, 12}

In[ ]:=
```
Reduce[x+y+x*y+y^2==13&&x>0&&y>0,{x,y},Integers]
Divisors[13]
```
Out[ ]= False

Out[ ]= {1, 13}

In[ ]:=
```
Reduce[x+y+x*y+y^2==14&&x>0&&y>0,{x,y},Integers]
Divisors[14]
```
Out[ ]= x == 6 && y == 1

Out[ ]= {1, 2, 7, 14}

In[]:=
```
Reduce[x+y+x*y+y^2==15&&x>0&&y>0,{x,y},Integers]
Divisors[15]
```

Out[]= x == 3 && y == 2

Out[]= {1, 3, 5, 15}

In[]:=
```
Reduce[x+y+x*y+y^2==16&&x>0&&y>0,{x,y},Integers]
Divisors[16]
```

Out[]= (x == 1 && y == 3) || (x == 7 && y == 1)

Out[]= {1, 2, 4, 8, 16}

In[]:=
```
Reduce[x+y+x*y+y^2==17&&x>0&&y>0,{x,y},Integers]
Divisors[17]
```

Out[]= False

Out[]= {1, 17}

In[]:=
```
Reduce[x+y+x*y+y^2==18&&x>0&&y>0,{x,y},Integers]
Divisors[18]
```

Out[]= (x == 4 && y == 2) || (x == 8 && y == 1)

Out[]= {1, 2, 3, 6, 9, 18}

In[]:=
```
Reduce[x+y+x*y+y^2==19&&x>0&&y>0,{x,y},Integers]
Divisors[19]
```

Out[]= False

Out[]= {1, 19}

In[]:=
```
Reduce[x+y+x*y+y^2==20&&x>0&&y>0,{x,y},Integers]
Divisors[20]
```

Out[]= (x == 2 && y == 3) || (x == 9 && y == 1)

Out[]= {1, 2, 4, 5, 10, 20}

In[]:=
```
Reduce[x+y+x*y+y^2==21&&x>0&&y>0,{x,y},Integers]
Divisors[21]
```

Out[]= x == 5 && y == 2

Out[]= {1, 3, 7, 21}

In[]:=
```
Reduce[x+y+x*y+y^2==22&&x>0&&y>0,{x,y},Integers]
Divisors[22]
```

Out[]= x == 10 && y == 1

Out[]= {1, 2, 11, 22}

In[]:=
```
Reduce[x+y+x*y+y^2==23&&x>0&&y>0,{x,y},Integers]
Divisors[23]
```

Out[]= False

Out[]= {1, 23}

In[ ]:=
```
Reduce[x+y+x*y+y^2==24&&x>0&&y>0,{x,y},Integers]
Divisors[24]
```

Out[ ]= (x == 3 && y == 3) || (x == 6 && y == 2) || (x == 11 && y == 1)

Out[ ]= {1, 2, 3, 4, 6, 8, 12, 24}

In[ ]:=
```
Reduce[x+y+x*y+y^2==25&&x>0&&y>0,{x,y},Integers]
Divisors[25]
```

Out[ ]= x == 1 && y == 4

Out[ ]= {1, 5, 25}

In[ ]:=
```
Reduce[x+y+x*y+y^2==26&&x>0&&y>0,{x,y},Integers]
Divisors[26]
```

Out[ ]= x == 12 && y == 1

Out[ ]= {1, 2, 13, 26}

In[ ]:=
```
Reduce[x+y+x*y+y^2==27&&x>0&&y>0,{x,y},Integers]
Divisors[27]
```

Out[ ]= x == 7 && y == 2

Out[ ]= {1, 3, 9, 27}

In[ ]:=
```
Reduce[x+y+x*y+y^2==28&&x>0&&y>0,{x,y},Integers]
Divisors[28]
```

Out[ ]= (x == 4 && y == 3) || (x == 13 && y == 1)

Out[ ]= {1, 2, 4, 7, 14, 28}

In[ ]:=
```
Reduce[x+y+x*y+y^2==29&&x>0&&y>0,{x,y},Integers]
Divisors[29]
```

Out[ ]= False

Out[ ]= {1, 29}

Main Results

We have found a way to identify prime numbers by an equation. **A number p is prime iff there are no positive solutions to diophantine equation x+y+x*y+x^2=p**

For the non prime numbers all the proper divisors are identified; let (x,y) a solution of diophantine equation x+y+x*y+x^2=n then x+y is a divisor of n.