Meccano hexagons

https://github.com/heptagons/meccano/hexa

1 Meccano hexagons

A meccano hexagon can be formed easily attaching equilateral triangles as small as one unit side. Also joining six unit bars and using double bars twice a rigid hexagon side 1 is obtained as shown in figure 1. Next we will find a way to make more interesting hexagons with diagonals with lengths not exact multiples of hexagon side.

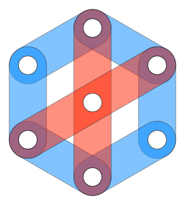

Figure 1: Rigid hexagon. Six blue bars form the perimeter with two double diagonals in red.

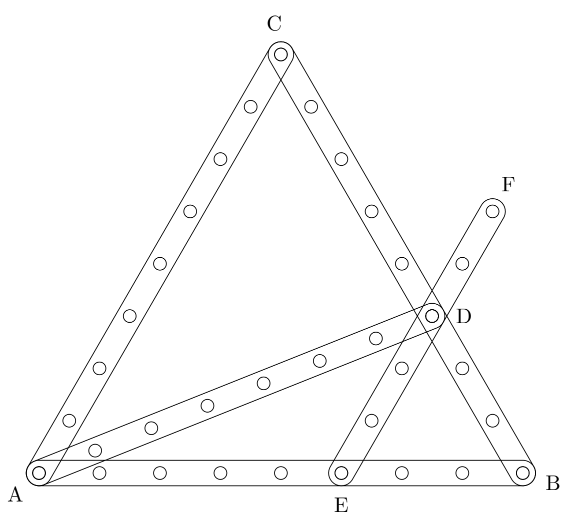

Figure 2: Regular hexagon internal angle plan. Consider an equilateral triangle ABC. Try to connect a new rod from point A to several points D located over the bar $\overline{BC}$ in such a way length $\overline{AD}$ is an integer. If so, connect a new rod $\overline{EF}$ where $\overline{EB} = \overline{BC}$ and $\overline{EF} = \overline{AE}$. Angle AEF is the internal regular hexagon angle (120°).

1.1 Independent diagonals

Figure 2 shows a plan to find independent diagonals that form the regular hexagon internal angles. From the figure the angle at B is fixed to $60°$, since ABC is a (rigid) equilateral triangle. Let a and b two integers such as $b <= a/2$.

We are looking for a third integer d to be the independent diagonal:

$$\begin{aligned}
a &= \overline{AB} \\
b &= \overline{BD} <= \frac{a}{2} \\
d &= \overline{AD} \\
e &= \overline{AE}
\end{aligned}$$

Acording to cosines law, the diagonal value is calculated as follows:

$$\begin{aligned}
d^2 &= a^2 + b^2 - 2ab\cos\frac{\pi}{3} \\
&= a^2 + b^2 - ab \\
&= (a-b)^2 + ab \\
&= e^2 + ab
\end{aligned}$$

Then, we need a program to iterate over integers a, then over integers b to inspect whether d value is as an integer too.

1.2 Search of integer diagonals

Next golang program find first cases. We iterate from $a = 1$ to a given maximum (line 2). Then we iterate from $b = 0$ to $b <= a/2$ (line 3). In order to reject repetitions by scaling we check for greatest commond divisor of a and b to be 1 (line 4). Then we calculate the diagonal using the plan's formula (line 5) and accept only the case when the diagonal is a square number (line 8).

```
1  func triangle_diagonals(max int) {
2    for a := 1; a < max; a++ {
3      for b := 1; b <= a/2; b++ {
4        if gcd(a, b) == 1 {
5          diag := (a-b)*(a-b) + a*b
6          cd := math.Sqrt(float64(diag))
7          d := int(cd)
8          if cd == float64(d) {
9            num := float64(diag + a*a - b*b)
10           den := 2.0 * cd * float64(a)
11           angle := 180*math.Acos(num/den)/math.Pi
12           fmt.Printf("a=%3d b=%3d d=%3d angle=%8.4f\n", a, b, d, angle)
13         }
14       }
15     }
16   }
17 }
18 func gcd(a, b int) int { // greatest common divisor
19   if b == 0 {
20     return a
21   }
```

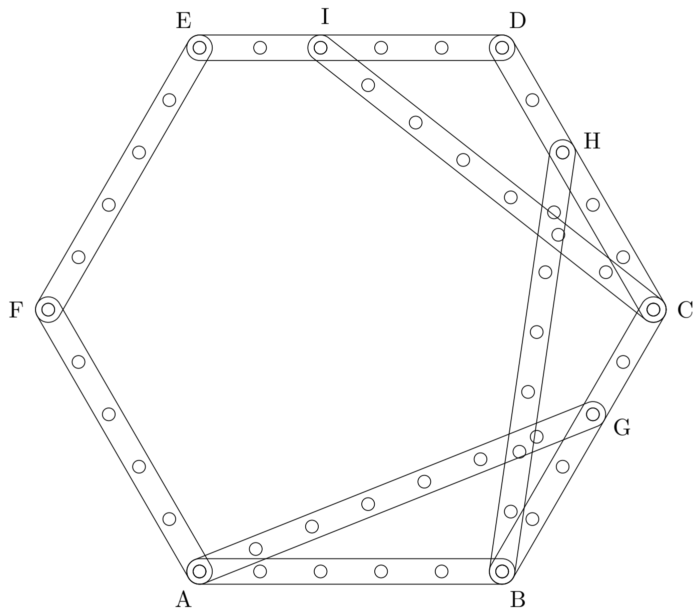

Figure 3: Hexagon sides 5, diagonals 7.

```
  return gcd(b, a % b)
}
```

1.3 Integral diagonals results

The program found 13 cases with integral diagonals for hexagons side <= 100. Each result includes lenghts a, b, d and the angle DAE (see figure 2).

```
a=  8 b=  3 d=  7 angle= 21.7868
a= 15 b=  7 d= 13 angle= 27.7958
a= 21 b=  5 d= 19 angle= 13.1736
a= 35 b= 11 d= 31 angle= 17.8966
a= 40 b=  7 d= 37 angle=  9.4300
a= 48 b= 13 d= 43 angle= 15.1782
a= 55 b= 16 d= 49 angle= 16.4264
a= 65 b=  9 d= 61 angle=  7.3410
a= 77 b= 32 d= 67 angle= 24.4327
a= 80 b= 17 d= 73 angle= 11.6351
a= 91 b= 40 d= 79 angle= 26.0078
a= 96 b= 11 d= 91 angle=  6.0090
a= 99 b= 19 d= 91 angle= 10.4174
```

1.4 Examples

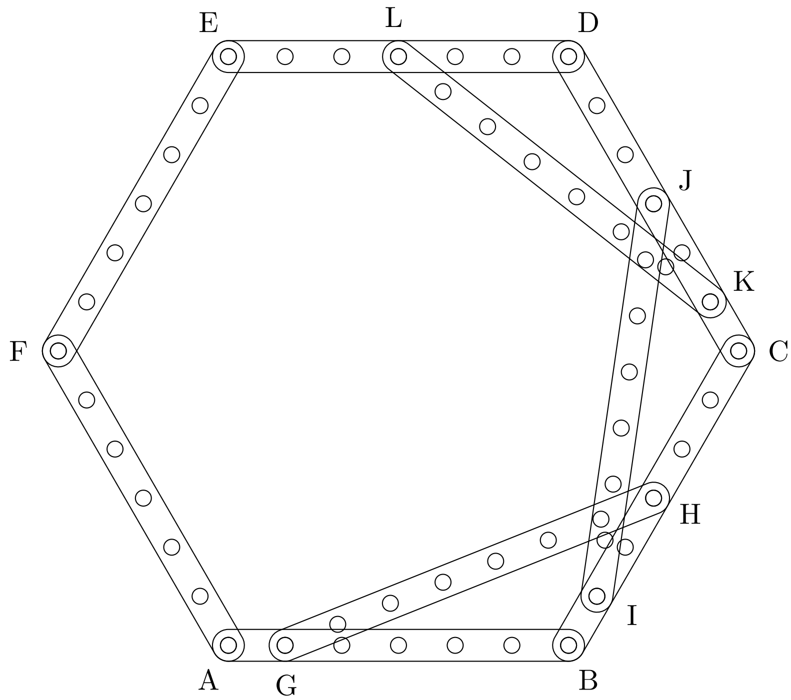

Figure 4: Hexagon sides $5 + 1 = 6$, diagonals 7.

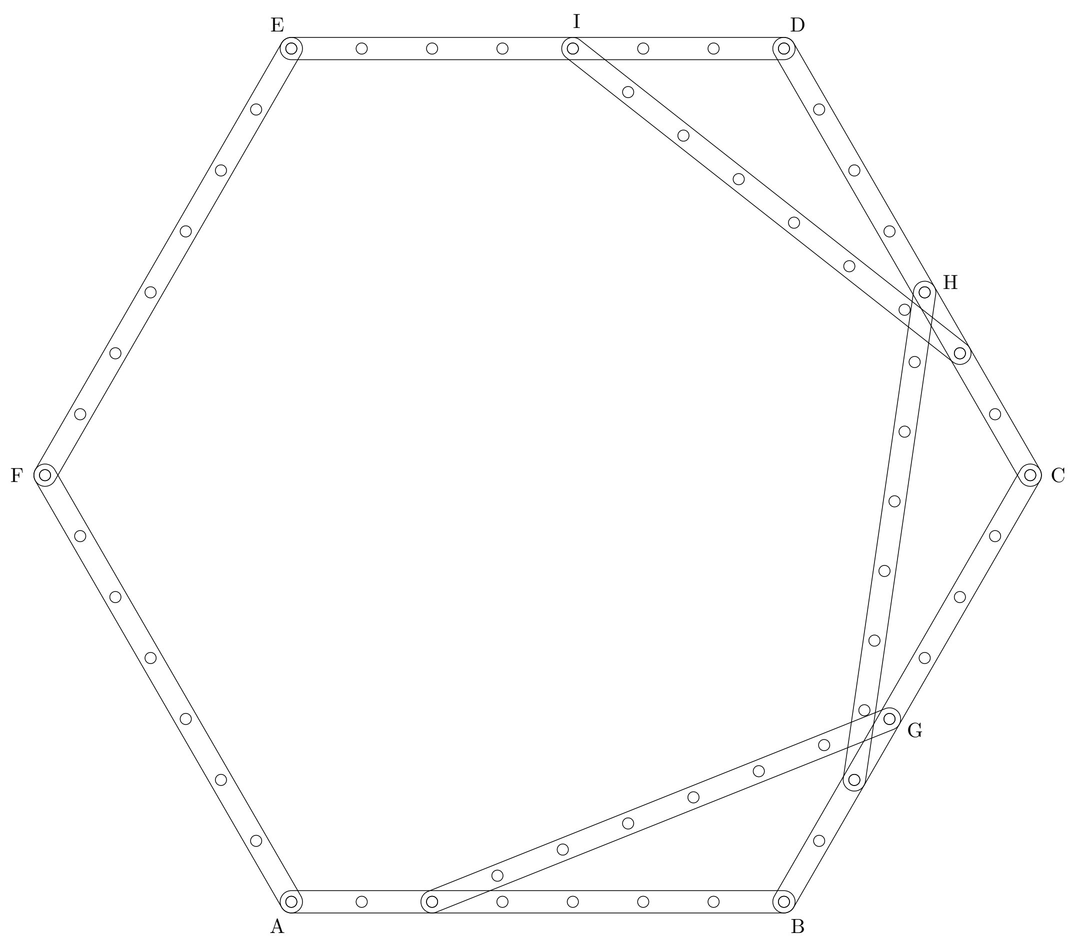

Figure 5: Hexagon sides $5 + 2 = 7$, diagonals 7.

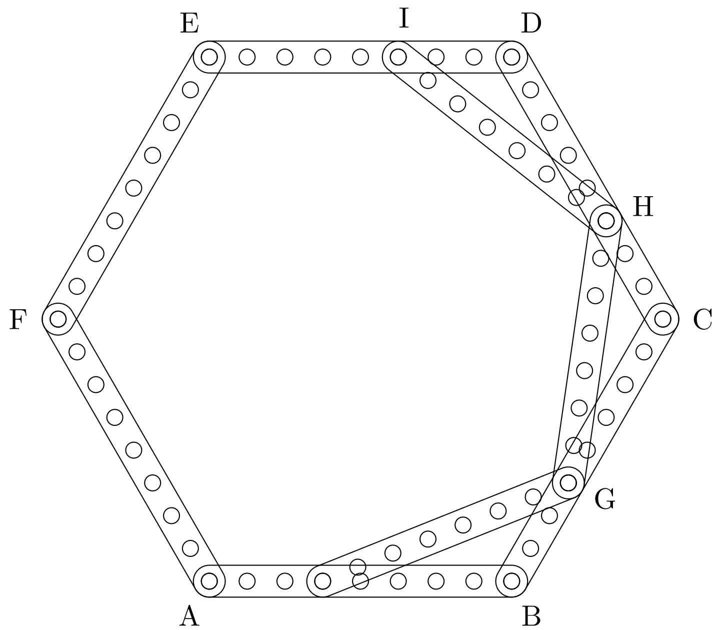

Figure 6: Hexagon sides $5 + 3 = 8$, diagonals 7.

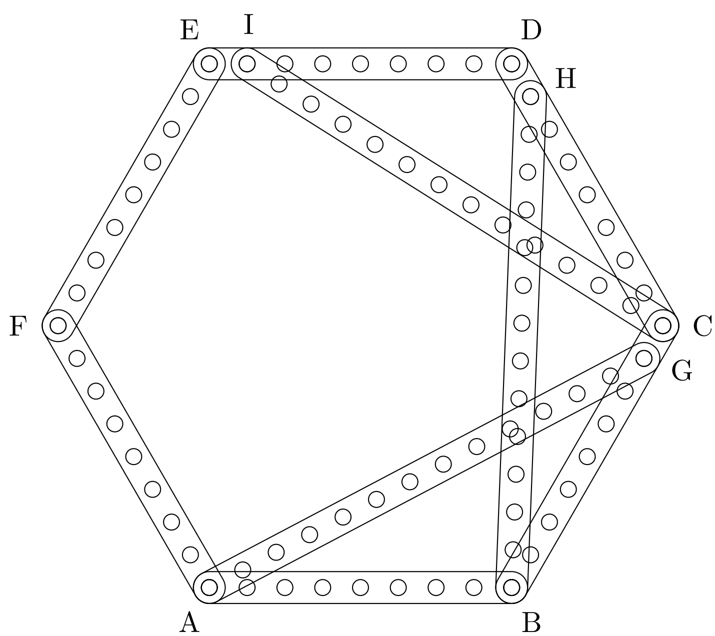

Figure 7: Hexagon sides 8, diagonals 13.