Meccano fox frame

https://github.com/heptagons/meccano/frames/fox

Abstract

Meccano fox frame is a group of five meccano [1] strips intended to be a base to build equilateral polygons. It resembles a fox face with two big pointing ears. The frame was used to build a regular pentagon[2] and here we explore more polygons. We conjecture the fox frame permits to build only a single pentagon, infinite hexagons, but no octagons, decagons nor dodecagons according to a brute-force search.

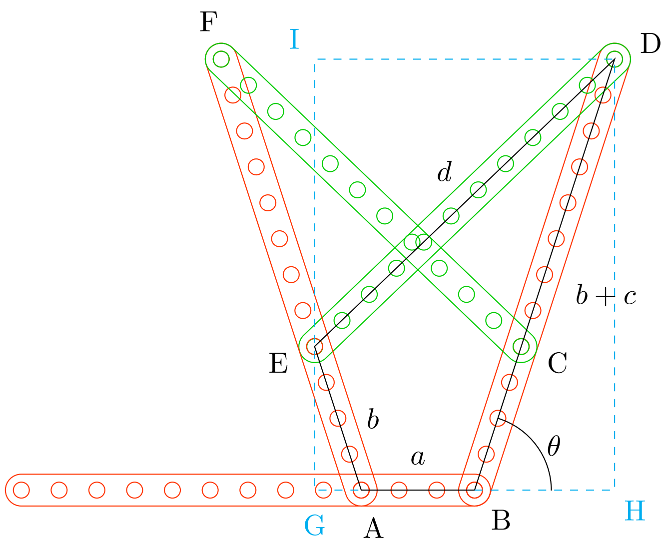

Figure 1: Fox-figure

Figure 1 show the so called meccano fox frame. Has five strips of three types:

- Single $\overline{AB}$ of length a.
- Pair { $\overline{BD}$, $\overline{AF}$ } of length $b+c$.
- Pair { $\overline{DE}$, $\overline{CF}$ } of length d.

In other words we have four different distances:

- a distance of segment $\overline{AB}$.
- b distance of segments $\overline{BC}$ and $\overline{AE}$.
- c distance of segments $\overline{CD}$ and $\overline{EF}$.
- d distance of segments $\overline{DE}$ and $\overline{CF}$.

We are going to test several values of (a, b, c, d) and calculate the angle $\angle HBD$. First we'll calculate a formula and then we'll run a program iterating integer values.

[1] Meccano mathematics by 't Hooft

[2] Meccano pentagons

1 Algebra

From figure 1 we define $\theta = \angle HBD$ and calculate sines and cosines:

$$\begin{aligned}
\theta &\equiv \angle HBD = \angle GAE && (1)\\
\overline{BH} &= (b+c)\cos\theta && (2)\\
\overline{DH} &= (b+c)\sin\theta && (3)\\
\overline{AG} &= b\cos\theta && (4)\\
\overline{EG} &= b\sin\theta && (5)
\end{aligned}$$

We calculate d in function of (a, b, c):

$$\begin{aligned}
d^2 &= (\overline{DE})^2\\
&= (\overline{DI})^2 + (\overline{EI})^2\\
&= (\overline{AG} + \overline{AB} + \overline{BH})^2 + (\overline{DH} - \overline{EG})^2 && (6)\\
&= (b\cos\theta + a + (b+c)\cos\theta)^2 + ((b+c)\sin\theta - b\sin\theta)^2 && (7)\\
&= (a + (2b+c)\cos\theta)^2 + (c\sin\theta)^2)\\
&= a^2 + 2a(2b+c)\cos\theta + (2b+c)^2\cos^2\theta + c^2\sin^2\theta\\
&= a^2 + 2a(2b+c)\cos\theta + (4b^2 + 4bc + c^2)\cos^2\theta + c^2\sin^2\theta\\
&= a^2 + 2a(2b+c)\cos\theta + (4b^2 + 4bc)\cos^2\theta + c^2\\
&= 4b(b+c)\cos^2\theta + 2a(2b+c)\cos\theta + a^2 + c^2 && (8)
\end{aligned}$$

We solve for $\cos\theta$ with the quadratic formula:

$$\begin{aligned}
\cos\theta &= \frac{-2a(2b+c) \pm \sqrt{4a^2(2b+c)^2 - 16b(b+c)(a^2+c^2-d^2)}}{8b(b+c)}\\
&= \frac{-a(2b+c) \pm \sqrt{a^2c^2 + 4b(b+c)(d^2-c^2)}}{4b(b+c)} && (9)
\end{aligned}$$

1.1 Test pentagon known case

Meccano fox frame appears in the single solution found of the meccano regular pentagon type 1 construction. In this case we have $a = 3$, $b = 4$, $c = 8$ and $d = 11$. Applying these values in the last equation we have:

$$\begin{aligned}
\cos\theta &= \frac{-48 \pm \sqrt{11520}}{192}\\
&= \frac{-1 \pm \sqrt{5}}{4} && (10)
\end{aligned}$$

Since $\cos 2\pi/5 = (\sqrt{5}-1)/4$ the equation for $\cos\theta$ passes the pentagon's test.

1.2 Meccano fox frame possible polygons

From figure 1 we notice angle $\angle ABC$ can be used as the internal angle of a regular polygon. The internal angle is the supplement of angle θ. Since $\cos\theta$ is an algebraic number of the form $\frac{B+C\sqrt{D}}{A}$ we can construct only a small group of regular polygons. Table 1 list such polygons excluding triangles and rectangles[3].

[3] Exact trigonometric values

Polygon	$\angle ABC$	θ	$\cos\theta$	$\{A, B, C, D\}$
Pentagon	$72°$	$\frac{2\pi}{5}$	$\frac{\sqrt{5}-1}{4}$	$\{4, -1, 1, 5\}$
Hexagon	$120°$	$\frac{\pi}{3}$	$\frac{1}{2}$	$\{2, 1, 0, 0\}$
Octagon	$135°$	$\frac{\pi}{4}$	$\frac{\sqrt{2}}{2}$	$\{2, 0, 1, 2\}$
Decagon	$144°$	$\frac{\pi}{5}$	$\frac{\sqrt{5}+1}{4}$	$\{4, 1, 1, 5\}$
Dodecagon	$150°$	$\frac{\pi}{6}$	$\frac{\sqrt{3}}{3}$	$\{3, 0, 1, 3\}$

Table 1: Regular polygons with $\cos\theta$ of the form $\frac{B+C\sqrt{D}}{A}$ where $A, D \in \mathbb{N}$ and $B, C \in \mathbb{Z}$.

2 Program

Next program iterates a, b, c, d to find polygons of different sizes. We set the maximum size to increase the strips lenghts and we get a callback with the sizes a, b, c, d and the algebraic cosine value of the form $\frac{B+C\sqrt{D}}{A}$. The algorithm prevents repetitions by scale. We use the package `github.com/heptagons/meccano/nest` algebra system.

```
func Fox(max N32, found func(a, b, c, d N32, cos *A32)) {
  factory := NewA32s()
  n1 := N32(1)
  for a := n1; a <= max; a++ {
    for b := n1; b <= max; b++ {
      ab := NatGCD(a, b)
      for c := n1; c <= max; c++ {
        abc := NatGCD(ab, c)
        na := N32(4)*b*(b+c)         // 4b(b+c)
        zb := -Z(a)*(2*Z(b) + Z(c)) // -a(2b+c)
        zc := Z(1)                   // 1
        a2c2 := Z(a*a)*Z(c*c)        // a
        for d := c; d <= max; d++ { // d >= c always
          if g := NatGCD(abc, d); g > 1 {
            continue // skip scale repetitions, eg. [1,2,3,4] = [2,4,6,8]
          }
          if zd := a2c2 + 4*Z(b)*Z(b+c)*(Z(d*d) - Z(c*c)); zd < 0 {
            // skip imaginary numbers invalid fox, like d too short
          } else if cos, err := factory.ANew3(N(na), zb, zc, zd); err != nil {
            // silent overflow
          } else {
            found(a, b, c, d, cos)
          }
        }
      }
    }
  }
}
```

2.1 Pentagons

As mentioned above, this program found only a single pentagon. We use this call:

```
func TestFoxPentagons(t *testing.T) {
  max := N32(100)
  fmt.Printf("max-lenght=%d a,b,c,d pentagons:\n", max)
  i := 0
  Fox(max, func(a, b, c, d N32, cos *A32) {
    if cos.Equals(4, -1, 1, 5) { //  cos 72
      i++
      fmt.Printf("% 3d %d,%d,%d,%d\n", i, a, b, c, d)
    }
  })
}
```

And we get the single result $a = 3, b = 4, c = 8, d = 11$:

```
max-lenght=100 a,b,c,d pentagons:
  1 3,4,8,11
```

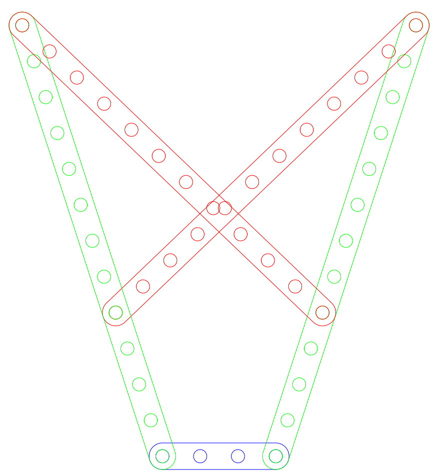

Figure 2: Meccano fox frame $a = 3, b = 4, c = 8, d = 11$. $\theta = 72°$.

2.2 Hexagons

More interesting costructions are the hexagons, since the algorithm found several. We run and filter the solutions where $\cos\theta = 1/2$. In order to build efficient hexagons we impose another condition $a > b + c$. This way the hexagons size will be a and the number of strips will be small since diagonals will remain inside each hexagon.

```
func TestFoxHexagons(t *testing.T) {
  max := N32(40)
  fmt.Printf("max-lenght=%d a,b,c,d efficient hexagons:\n", max)
  i := 0
  Fox(max, func(a, b, c, d N32, cos *A32) {
    if cos.Equals(2,1) { //  cos 60
      // Efficient hexagons are those when a > b+c
      if a >= b+c {
        i++
        fmt.Printf("% 3d %d,%d,%d,%d\n", i, a, b, c, d)
      }
```

```
    }
  })
}
```

We found 42 different hexagons when the maximum strip is of size 40 as shown in next table. Each row last four numbers correspond to the lenghts of the strips segments a, b, c, d:

```
  1 4,1,3,7
  2 9,1,6,14
  3 11,5,5,19
  4 12,4,5,19
  5 13,2,9,21
  6 13,3,5,19
  7 14,1,9,21
  8 14,2,5,19
  9 15,1,5,19
 10 15,1,14,26
 11 17,3,12,28
 12 18,6,11,31
 13 19,1,12,28
 14 19,5,11,31
 15 20,4,11,31
 16 20,13,7,37
 17 21,3,11,31
 18 21,4,15,35
 19 21,11,10,38
 20 21,12,7,37
 21 22,2,11,31
 22 22,3,15,35
 23 22,11,7,37
 24 23,1,11,31
 25 23,1,21,39
 26 23,2,15,35
 27 23,9,10,38
 28 23,10,7,37
 29 24,1,15,35
 30 24,9,7,37
 31 25,7,10,38
 32 25,8,7,37
 33 26,7,7,37
 34 27,5,10,38
 35 27,6,7,37
 36 28,5,7,37
 37 29,3,10,38
 38 29,4,7,37
 39 30,3,7,37
 40 31,1,10,38
 41 31,2,7,37
 42 32,1,7,37
```

When we extend the maximum size to 100 we found 350 hexagons where the last one has strips $a = 84, b = 1, c = 11, d = 91$.

2.3 Octagons and dodecagons

The program found no octagons nor dodecagons by calling these two functions (time ellapsed around 116 seconds each):

```
func TestFoxOctagons(t *testing.T) {
  max := N32(80)
  fmt.Printf("max-lenght=%d a,b,c,d octagons:\n", max)
  i := 0
  Fox(max, func(a, b, c, d N32, cos *A32) {
    if cos.Equals(2,0,1,2) { //  cos 45 degrees sqrt{2}/2
      i++
      fmt.Printf("% 3d %d,%d,%d,%d\n", i, a, b, c, d)
    }
  })
}

func TestFoxDodecagons(t *testing.T) {
  max := N32(80)
  fmt.Printf("max-lenght=%d a,b,c,d dodecagons:\n", max)
  i := 0
  Fox(max, func(a, b, c, d N32, cos *A32) {
    if cos.Equals(3,0,1,3) { //  cos 30 degrees sqrt{3}/3
      i++
```

```
            fmt.Printf("% 3d %d,%d,%d,%d\n", i, a, b, c, d)
        }
    })
}
```

3 Conjectures

According to the program results, we conjecture that including the **fox frame** exist: A single pentagon, infinite hexagons, zero octagons, zero decagons and zero dodecagons.

4 Fox hexagons examples

Here we build the first hexagons from the list values. The figures are not presented originally in the paper Meccano hexagons[4]. In that paper the so-called irregular diagonals connects hexagon's adjacent sides. The difference here is that fox diagonals d connects not adjacent sides but skips one. From figure 1 we see the irregular diagonal d connects hexagon's side $\overline{FA}$ with side $\overline{BD}$ skipping side $\overline{AB}$.

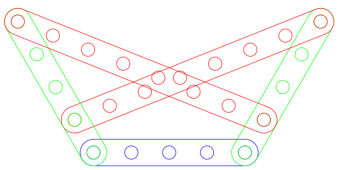

Figure 3: Meccano fox(4,1,3,7), $\theta = 60°$.

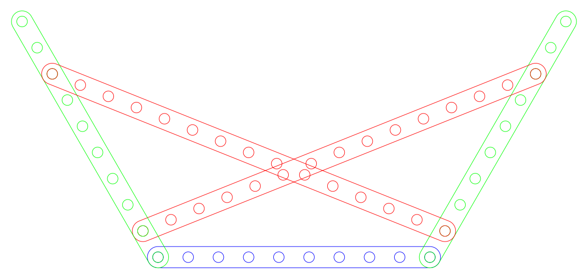

Figure 4: Meccano fox(9,1,6,14), $\theta = 60°$.

Figures 3 and 4 show the two first hexagon fox frames. With two copies of each unit we can build a complete hexagon, so we'll have 10 strips. The second copy is rotated 180° and the foxes ears get in touch, imagine that. Also, we can remove a red diagonal and such hexagon remains rigid.

Also we can build the hexagon with 9 pieces with three-fold symmetry. We start taking only three strips from each unit, namely a semi-unit including the blue strip, one red and one green as seen in the figures. Then the semi-unit is cloned two times and rotated each 120°. Last figures show these reduced and reinforced constructions.

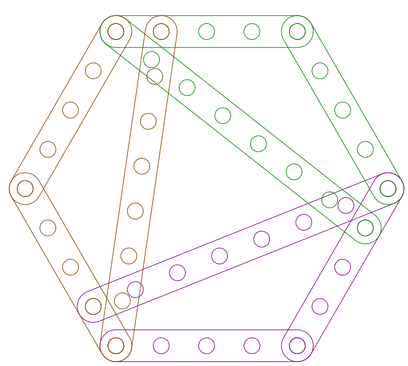

Figure 5: Hexagon side 4 with three fox semi-frames (4,1,3,7).

[4] Meccano hexagons

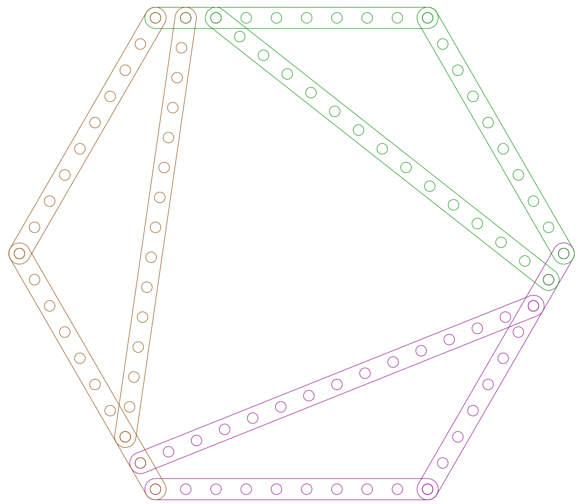

Figure 6: Hexagon side 9 with three fox semi-frames (9,1,6,14).

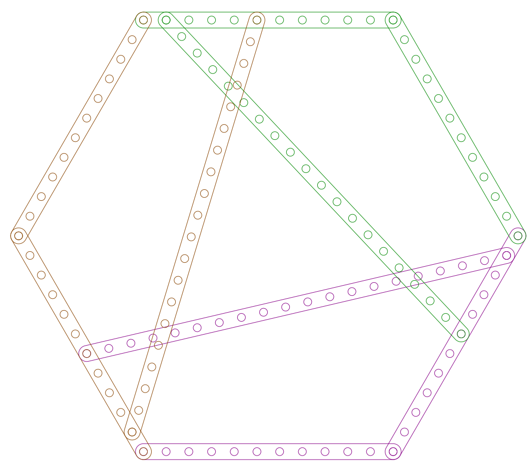

Figure 7: Hexagon side 11 with three fox semi-frames (11,5,5,19).

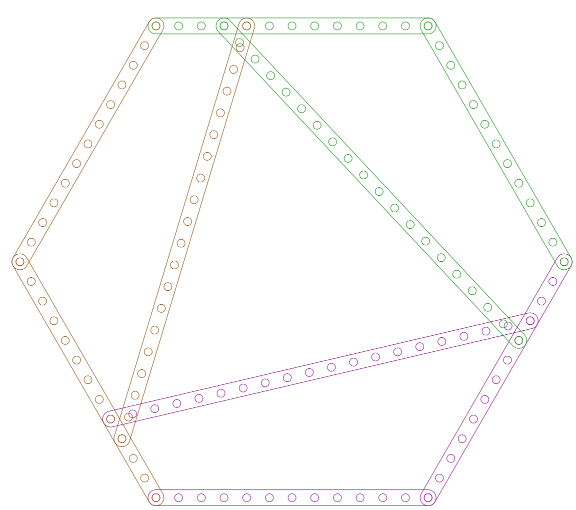

Figure 8: Hexagon side 12 with three fox semi-frames (12,4,5,19).