

```
In[1]:= Needs["Combinatorica`"]
```

■ Graph theory routines

```
In[2]:= "Substitutes for defunct routines in the Combinatorica package";
```

```
In[3]:= makegraph[n_, nbrfn_] := Table[If[nbrfn[i, j], 1, 0], {i, n}, {j, n}]
```

```
In[4]:= "Breadth-first search";
```

```
bfs[g_, a_] := Module[{n, scan, queue, i},
  n = Length[g];
  scan = {a};
  queue = {a};
  While[queue != {},
    i = queue[[1]];
    Do[If[g[[i, j]] == 1 && Not[MemberQ[scan, j]],
      scan = Append[scan, j];
      queue = Append[queue, j]
    ],
    {j, 1, n}
  ];
  queue = Drop[queue, 1]
];
scan
```

■ General Definitions

```

In[6]:= DrumNbr[d_, i_, s_] := d[[s, i]];
DrumAdjacent[d_, i_, j_] :=
  DrumNbr[d, i, 1] == j || DrumNbr[d, i, 2] == j || DrumNbr[d, i, 3] == j;

DrumGraph[d_] := makegraph[Length[d[[1]]], DrumAdjacent[d, #1, #2] &];

DrumDomain[d_, t_, start_] :=
  Module[{n = Length[d[[1]]], scanorder, regions, i, j, k, side}, regions = Table[{}, {n}];
  regions[[start]] = t;
  scanorder = bfs[DrumGraph[d], start];
  Do[i = scanorder[[k]];
    j = DrumNbr[d, i, side];
    If[regions[[i]] != {} && regions[[j]] == {}, regions[[j]] = Flip[regions[[i]], side]],
    {k, 1, Length[scanorder]}, {side, 1, 3}];
  {d, regions}];

Triangle[a_, b_, c_] := Module[{x, y, z}, x = Pi / 2 (b + c - a) / (a + b + c);
  y = Pi / 2 (a + c - b) / (a + b + c);
  z = Pi / 2 (a + b - c) / (a + b + c);
  Map[N[{Cos[#], Sin[#]}] &, {-z, Pi + z, Pi / 2 + x - y}]];

Flip[{a_, b_, c_}] := Module[{x = a - c, y = b - c}, {c + (-1) x + 2 (x.y) / (y.y) y, b, c}];
Flip[t_, n_] := RotateLeft[Flip[RotateLeft[t, n - 1]], -(n - 1)];

DrumSideStyle0 = Module[{int = AbsoluteThickness[.5], ext = AbsoluteThickness[3],
  side1 = AbsoluteDashing[{}], side2 = AbsoluteDashing[{}], side3 = AbsoluteDashing[{}],
  {{int, side1}, {int, side2}, {int, side3}}, {{ext, side1}, {ext, side2}, {ext, side3}}];
DrumSideStyle1 = Module[{int = AbsoluteThickness[.5], ext = AbsoluteThickness[3], side1 =
  AbsoluteDashing[{1, 5}], side2 = AbsoluteDashing[{10, 10}], side3 = AbsoluteDashing[{}],
  {{int, side1}, {int, side2}, {int, side3}}, {{ext, side1}, {ext, side2}, {ext, side3}}];
DrumSideStyle = DrumSideStyle1;

DrumTriangle[t_, label_, exterior_] :=
  {GrayLevel[1.], Polygon[t], GrayLevel[0], Text[label, (t[[1]] + t[[2]] + t[[3]]) / 3,
  DrumSideStyle[If[exterior[[1]], 2, 1], 1] ~Join~ {Line[{t[[2]], t[[3]]}]},
  DrumSideStyle[If[exterior[[2]], 2, 1], 2] ~Join~ {Line[{t[[1]], t[[3]]}]},
  DrumSideStyle[If[exterior[[3]], 2, 1], 3] ~Join~ {Line[{t[[1]], t[[2]]}]}];

DrumDomainGraphics[{d_, regions_}, label_] :=
  Module[{i, s}, Graphics[Table[DrumTriangle[regions[[i]], label[i - 1],
  Table[DrumNbr[d, i, s] == i, {s, 3}]], {i, Length[regions]}], AspectRatio -> Automatic]];

StretchOut[s1_, s2_] := {s1[[1]], s1[[1]] + Max[s1[[2]] - s1[[1]], s2[[2]] - s2[[1]]]};
StretchOut[{a1_, a2_}, {b1_, b2_}] := Module[{border = .02, size},
  size = Max[a2 - a1, b2 - b1];

```

```

    {a1 - border * size, a1 + size + border * size}
]
StretchRange[r1_, r2_] := {StretchOut[r1[[1]], r2[[1]]], StretchOut[r1[[2]], r2[[2]]]}
ExpandRange[g1_, g2_] := StretchRange[FullOptions[g1, PlotRange], FullOptions[g2, PlotRange]]

DrumLabel0[i_] := "";
DrumLabel1[i_] := i;
DrumLabel2[i_] := If[i == 0, "0", ""];
DrumLabel = DrumLabel0;
DrumLabelLeft := DrumLabel;
DrumLabelRight := DrumLabel;

DrumPairGraphics[dd_, t_, {start1_, start2_}] := Module[{g1, g2, gg, g1new, g2new, p},
  g1 = DrumDomainGraphics[DrumDomain[dd[[1]], t, start1 + 1], DrumLabelLeft];
  g2 = DrumDomainGraphics[DrumDomain[dd[[2]], t, start2 + 1], DrumLabelRight];
  g1new = Show[g1, PlotRange → ExpandRange[g1, g2], DisplayFunction → Identity];
  g2new = Show[g2, PlotRange → ExpandRange[g2, g1], DisplayFunction → Identity];
  {FullOptions[g2, PlotRange], FullOptions[g2new, PlotRange]};
  GraphicsArray[{g1new, g2new}]];

DrumPair[dd_, a_, b_, c_, start_ : {0, 0}] := DrumPairGraphics[dd, Triangle[a, b, c], start];

ShowDrumPair[dd_, a_, b_, c_, start_] := Show[DrumPair[dd, a, b, c, start]];
ShowDrumPair[dd_, a_, b_, c_] := ShowDrumPair[dd, a, b, c, {0, 0}];
ShowDrumPair[dd_, start_] := ShowDrumPair[dd, 60, 60, 60, start];
ShowDrumPair[dd_] := ShowDrumPair[dd, 60, 60, 60, {0, 0}];

ShowLabel[g_, label_] :=
  Show[Graphics[{Rectangle[{0, 0}, {1, 1}], g}, Text[label, {.5, -.1}], Text["", {.5, -.2}]]];

FromCyclesNum[cyc_List, n_Integer] := Module[{p = Range[n], pos}, Scan[(pos = Last[#];
  Scan[Function[c, pos = p[[pos]] = c], #]) &, cyc];
  p];

ConjugatePermutation[p_?PermutationQ, q_?PermutationQ] := Permute[q, Permute[p, q]];
Conj = ConjugatePermutation;

ConjugatePermutation2[p_, q_] := Thread[ConjugatePermutation[p, q]];
Conj2 = ConjugatePermutation2;

C2[a_, b_] := ConjugatePermutation2[a, b];
C2[a_, b_, c_] := C2[a, C2[b, c]];
C2[a_, b_, c_, d_] := C2[a, C2[b, c, d]];
C2[a_, b_, c_, d_, e_] := C2[a, C2[b, c, d, e]];

LeftBraid[{a_, b_, c_}] := {ConjugatePermutation[b, a], a, c};
RightBraid[{a_, b_, c_}] := {a, c, ConjugatePermutation[b, c]};
UnLeftBraid[{a_, b_, c_}] := {b, ConjugatePermutation[a, b], c};

```

```

UnRightBraid[{a_, b_, c_}] := {a, ConjugatePermutation[c, b], b};

DrumLeft[d_] := Map[LeftBraid, d];
DrumRight[d_] := Map[RightBraid, d];
DrumUnLeft[d_] := Map[UnLeftBraid, d];
DrumUnRight[d_] := Map[UnRightBraid, d];

Prep[l_, n_] := FromCyclesNum[l + 1, n];

Drum2[a_, b_, c_] := Transpose[{a, b, c}];

PermutationMatrix[p_?PermutationQ] := Permute[IdentityMatrix[Length[p]], p];
DrumAdjacencyMatrix[d_List] :=
  a PermutationMatrix[d[[1]]] + b PermutationMatrix[d[[2]]] + c PermutationMatrix[d[[3]]];
NeumannMatrix[d_List] := (a + b + c) IdentityMatrix[Length[d[[1]]]] - DrumAdjacencyMatrix[d];
DirichletMatrix[d_List] :=
  (a + b + c) IdentityMatrix[Length[d[[1]]]] + NegateOffDiagonal[DrumAdjacencyMatrix[d]];
OffDiagonal[m_] := (1 - IdentityMatrix[Length[m]]) m;
NegateOffDiagonal[m_] := m - 2 OffDiagonal[m];
NeumannPolynomial[d_List] := CharacteristicPolynomial[NeumannMatrix[d], x];
DirichletPolynomial[d_List] := CharacteristicPolynomial[DirichletMatrix[d], x];
DrumPairTest[p_] := {NeumannPolynomial[p[[1]]] == NeumannPolynomial[p[[2]]],
  DirichletPolynomial[p[[1]]] == DirichletPolynomial[p[[2]]]};

```

■ Preliminary Definitions

```

In[63]:= Module[{a1, a2, b1, b2, c1, c2, d1, d2, e1, e2, f1, f2, g1, g2, h1, h2, i1, i2, j1, j2, k1, k2,
  l1, l2, p1, p2, q1, q2, r1, r2, a, b, c, d, e, f, g, h, i, j, k, l, p, q, r, aa, bb, cc,
  dd, ee, ff, gg, hh, ii, jj, kk, ll, pp, qq, rr, eee}, a1 = Prep[{0, 1}, {2, 5}, 7];
a2 = Prep[{0, 4}, {2, 3}, 7];
b1 = Prep[{0, 2}, {4, 3}, 7];
b2 = Prep[{0, 1}, {4, 6}, 7];
c1 = Prep[{0, 4}, {1, 6}, 7];
c2 = Prep[{0, 2}, {1, 5}, 7];
a = {a1, a2};
b = {b1, b2};
c = {c1, c2};
aa = Conj2[a, c];
bb = Conj2[b, a];
d1 = Prep[{0, 12}, {1, 10}, {3, 5}, {6, 7}, 13];
d2 = Prep[{0, 4}, {2, 3}, {6, 8}, {9, 10}, 13];
e1 = Prep[{0, 10}, {3, 4}, {9, 2}, {5, 8}, 13];
e2 = Prep[{0, 12}, {6, 9}, {5, 11}, {1, 4}, 13];
f1 = Prep[{0, 4}, {9, 12}, {1, 6}, {2, 11}, 13];
f2 = Prep[{0, 10}, {5, 1}, {2, 7}, {3, 12}, 13];
d = {d1, d2};
e = {e1, e2};
f = {f1, f2};
ee = Conj2[e, d];
dd = Conj2[d, f];

```

```

ff = Conj2[f, ee];
eee = Conj2[ee, dd];
g1 = Prep[{0, 2}, {1, 7}, {3, 6}, {5, 10}}, 13];
g2 = Prep[{0, 7}, {3, 11}, {6, 8}, {9, 12}}, 13];
h1 = Prep[{0, 6}, {3, 8}, {9, 5}, {2, 4}}, 13];
h2 = Prep[{0, 8}, {9, 7}, {5, 11}, {1, 10}}, 13];
i1 = Prep[{0, 5}, {9, 11}, {1, 2}, {6, 12}}, 13];
i2 = Prep[{0, 11}, {1, 8}, {2, 7}, {3, 4}}, 13];
g = {g1, g2};
h = {h1, h2};
i = {i1, i2};
hh = Conj2[h, g];
gg = Conj2[g, i];
ii = Conj2[i, gg];
j1 = Prep[{0, 14}, {4, 5}, {9, 10}, {1, 12}, {7, 11}, {2, 6}}, 15];
j2 = Prep[{0, 11}, {1, 5}, {3, 4}, {6, 10}, {8, 9}, {13, 14}}, 15];
k1 = Prep[{4, 6}, {1, 13}, {8, 9}, {2, 7}}, 15];
k2 = Prep[{0, 10}, {1, 2}, {6, 9}, {12, 14}}, 15];
l1 = Prep[{14, 1}, {3, 4}, {12, 2}, {8, 11}}, 15];
l2 = Prep[{0, 5}, {2, 4}, {6, 7}, {11, 14}}, 15];
j = {j1, j2};
k = {k1, k2};
l = {l1, l2};
ll = Conj2[l, j];
jj = Conj2[j, k];
kk = Conj2[k, ll];
p1 = Prep[{2, 7}, {3, 11}, {5, 12}, {8, 18}, {13, 14}, {15, 17}, {16, 20}}, 21];
p2 = Prep[{0, 1}, {4, 17}, {7, 12}, {9, 16}, {10, 20}, {11, 13}, {15, 19}}, 21];
q1 = Prep[{0, 17}, {3, 8}, {4, 12}, {6, 13}, {9, 19}, {14, 15}, {16, 18}}, 21];
q2 = Prep[{0, 20}, {3, 16}, {6, 11}, {8, 15}, {9, 19}, {10, 12}, {14, 18}}, 21];
r1 = Prep[{1, 8}, {2, 16}, {4, 11}, {5, 19}, {7, 14}, {10, 17}, {13, 20}}, 21];
r2 = r1;
p = {p1, p2};
q = {q1, q2};
r = {r1, r2};
D71 = Drum2[a, b, c];
D72 = Drum2[a, bb, c];
D73 = Drum2[aa, bb, c];
D131 = Drum2[d, e, f];
D132 = Drum2[d, ee, f];
D133 = Drum2[dd, ee, f];
D134 = Drum2[dd, ee, ff];
D135 = Drum2[dd, eee, ff];
D136 = Drum2[g, h, i];
D137 = Drum2[g, hh, i];
D138 = Drum2[gg, hh, i];
D139 = Drum2[gg, hh, ii];
D151 = Drum2[j, k, l];
D152 = Drum2[j, k, ll];
D153 = Drum2[jj, k, ll];

```

```
D154 = Drum2[jj, kk, ll];
D211 = Drum2[p, q, r];]
```

```
In[64]:= Clear[ShowFig0, ShowFig];
```

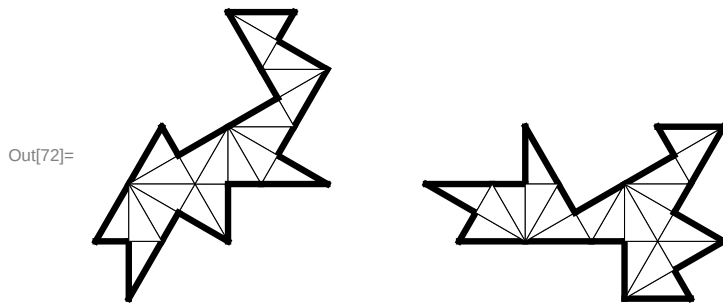
```
Fig0[d_, a_, b_, c_] :=
  Block[{DrumLabel = DrumLabel0, DrumSideStyle = DrumSideStyle0}, DrumPair[d, a, b, c]];
Fig0[d_] := Fig0[d, 60, 60, 60];
ShowFig0[d_, a_, b_, c_] := Show[Fig0[d, a, b, c]];
ShowFig0[d_] := Show[Fig0[d, 60, 60, 60]];

Fig[d_, a_, b_, c_] :=
  Block[{DrumLabel = DrumLabel2, DrumSideStyle = DrumSideStyle0}, DrumPair[d, a, b, c]];
Fig[d_] := Fig[d, 60, 60, 60];
```

■ Figures

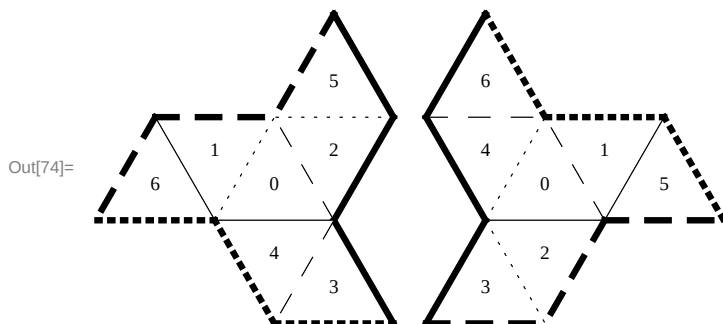
```
In[71]:= "Figure homophonic";
```

```
Show[Fig0[D211, 90, 60, 30]]
```



```
In[73]:= "Figure propeller";
```

```
Show[
  Block[{DrumLabel = DrumLabel1, DrumSideStyle = DrumSideStyle1}, DrumPair[D71, 60, 60, 60]]]
```

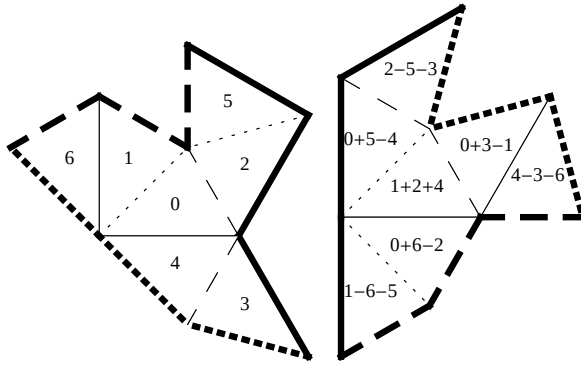


In[75]:= "Figure warped";

```
Fig4Label1[i_] := {"0", "1", "2", "3", "4", "5", "6"}[[i + 1]];
Fig4Label2[i_] := {"1+2+4", "0+3-1", "0+6-2", "1-6-5", "0+5-4", "4-3-6", "2-5-3"}[[i + 1]];

Block[{DrumLabelLeft = Fig4Label1, DrumLabelRight = Fig4Label2,
  DrumSideStyle = DrumSideStyle1}, Show[DrumPair[D71, 60, 45, 75]]]
```

Out[78]=



In[79]:= "Figure examples";

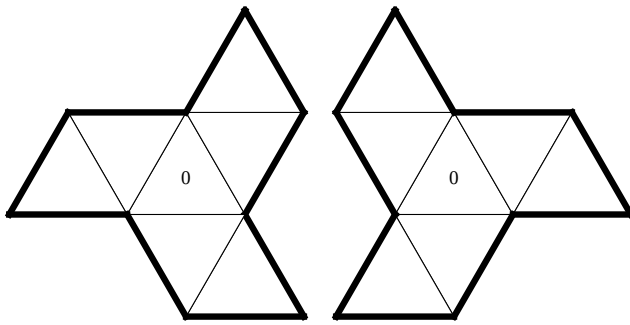
```
In[80]:= "Individual diagrams.";

F71 = ShowLabel[Fig[D71], Subscripted[7[1]]]
F72 = ShowLabel[Fig[D72], Subscripted[7[2]]]
F73 = ShowLabel[Fig[D73], Subscripted[7[3]]]

F131 = ShowLabel[Fig[D131], Subscripted[13[1]]]
F132 = ShowLabel[Fig[D132], Subscripted[13[2]]]
F133 = ShowLabel[Fig[D133], Subscripted[13[3]]]
F134 = ShowLabel[Fig[D134], Subscripted[13[4]]]
F135 = ShowLabel[Fig[D135], Subscripted[13[5]]]
F136 = ShowLabel[Fig[D136], Subscripted[13[6]]]
F137 = ShowLabel[Fig[D137], Subscripted[13[7]]]
F138 = ShowLabel[Fig[D138], Subscripted[13[8]]]
F139 = ShowLabel[Fig[D139], Subscripted[13[9]]]
F151 = ShowLabel[Fig[D151], Subscripted[15[1]]]
F152 = ShowLabel[Fig[D152], Subscripted[15[2]]]
F153 = ShowLabel[Fig[D153], Subscripted[15[3]]]
F154 = ShowLabel[Fig[D154], Subscripted[15[4]]]

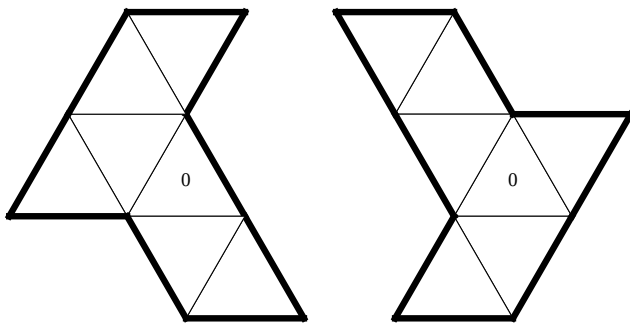
F211 = ShowLabel[Fig[D211], Subscripted[21[1]]]
```


Out[81]=



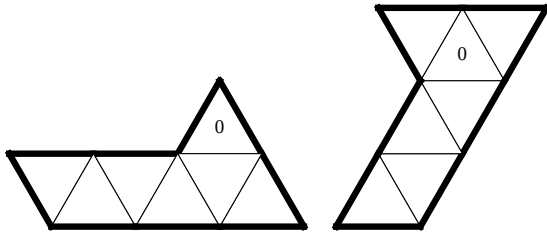
7_1

Out[82]=



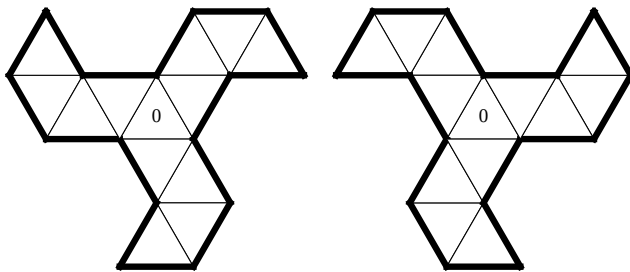
7_2

Out[83]=

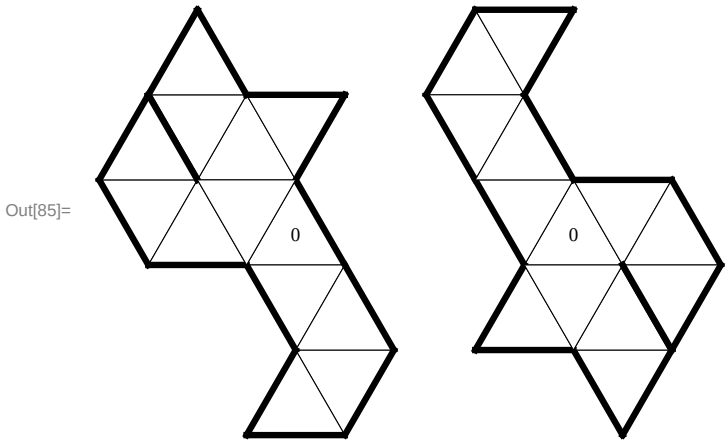


7_3

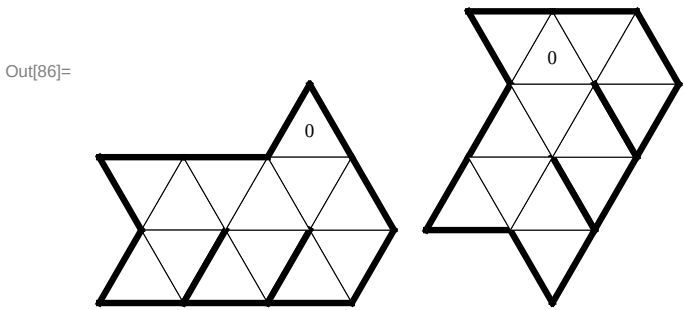
Out[84]=



13_1

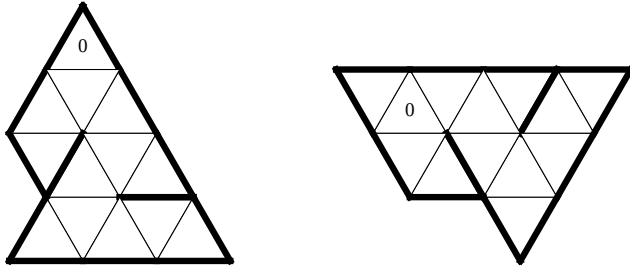


13_2



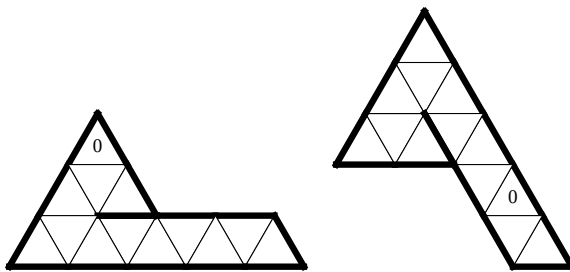
13_3

Out[87]=



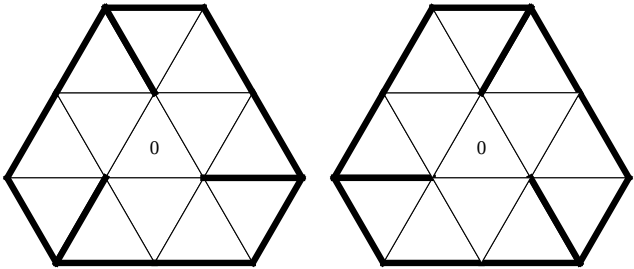
13_4

Out[88]=



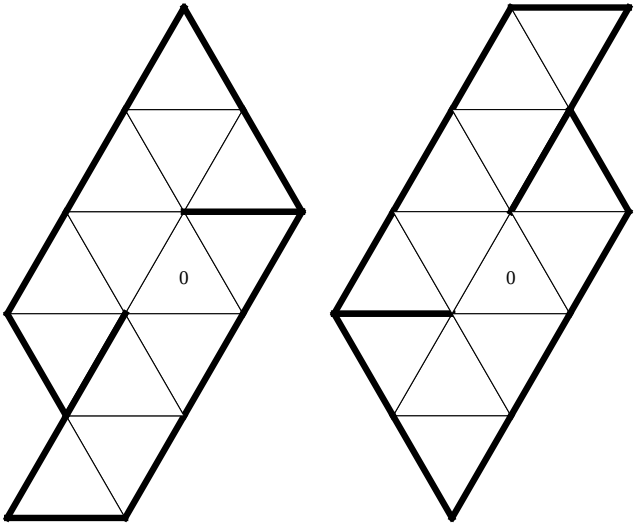
13_5

Out[89]=



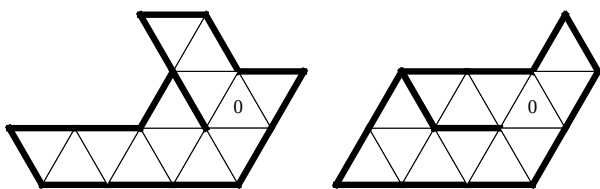
13₆

Out[90]=



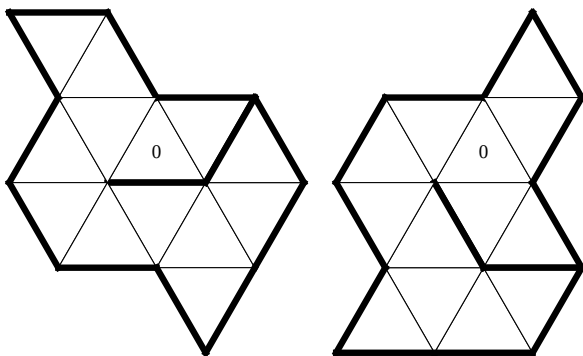
13₇

Out[91]=



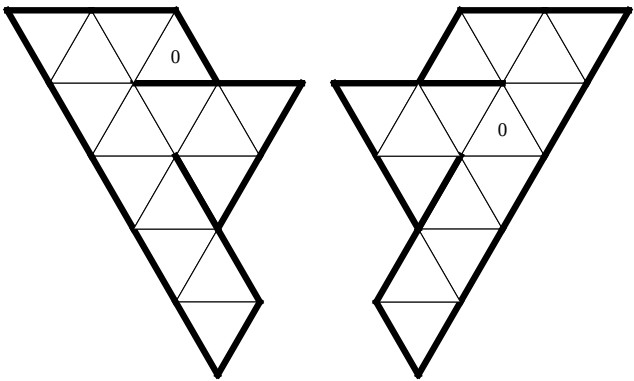
13_8

Out[92]=



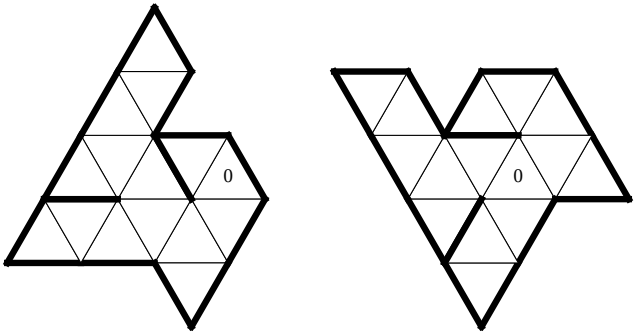
13_9

Out[93]=



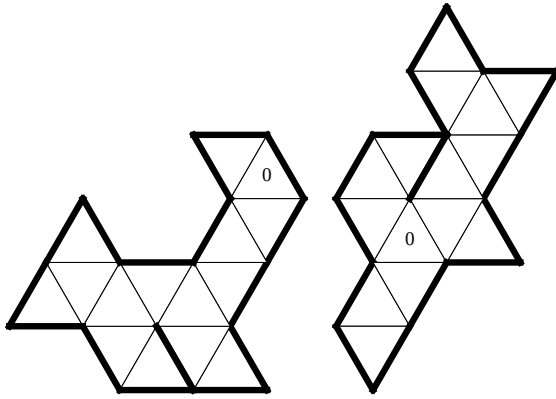
15₁

Out[94]=



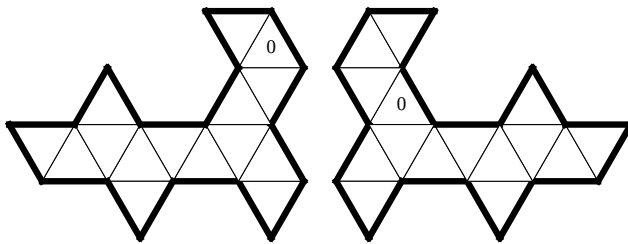
15₂

Out[95]=



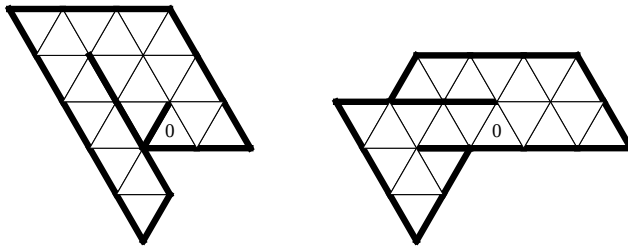
15_3

Out[96]=



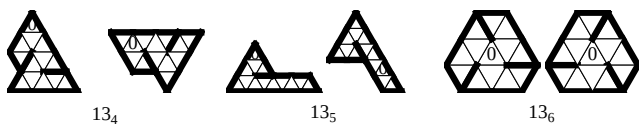
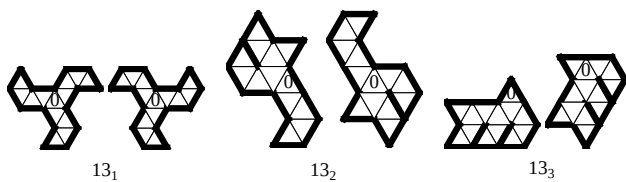
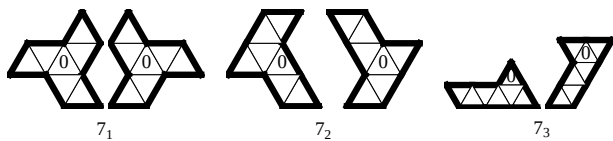
15_4

Out[97]=

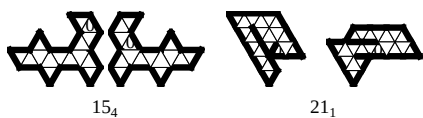
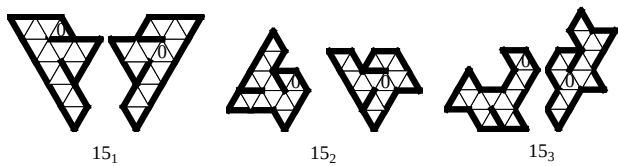
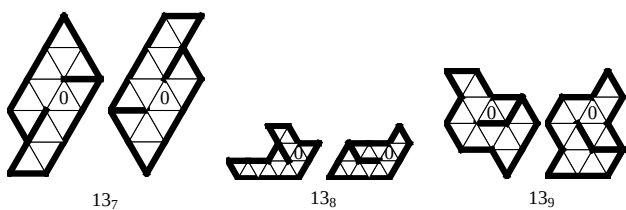
 21_1

In[98]:= "The figure itself.;"

```
Show[GraphicsArray[{{F71, F72, F73}, {F131, F132, F133},
  {F134, F135, F136}, {F137, F138, F139}, {F151, F152, F153}, {F154, F211}}]]
```



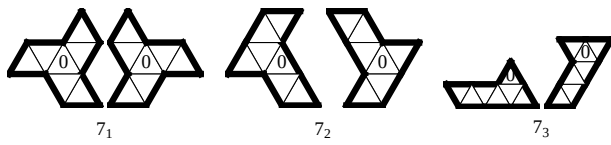
Out[99]=



```
In[100]:= "Examples from  $\text{PSL}(3,2)$  .";
```

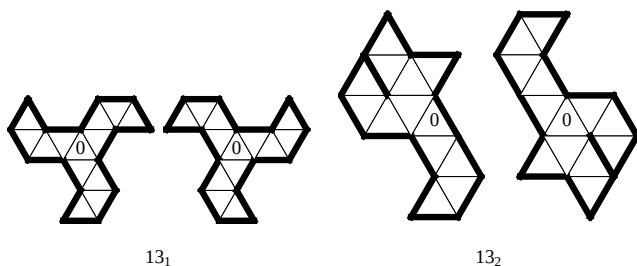
```
F7 = Show[GraphicsArray[{F71, F72, F73}]]
```

```
Out[101]=
```

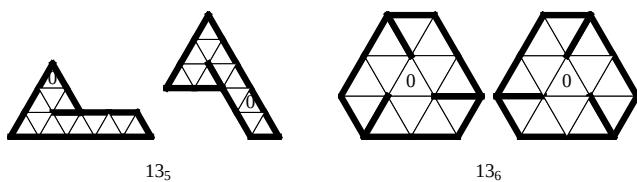
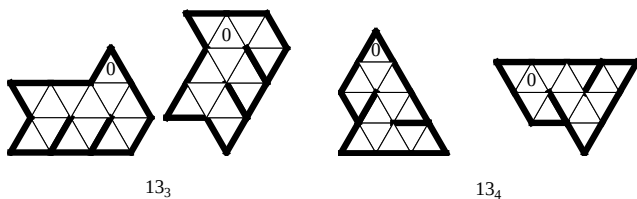


```
In[102]:= "Examples from PSL(3,3).";
```

```
F13 = Show[GraphicsArray[{{F131, F132}, {F133, F134}, {F135, F136}}]]
```

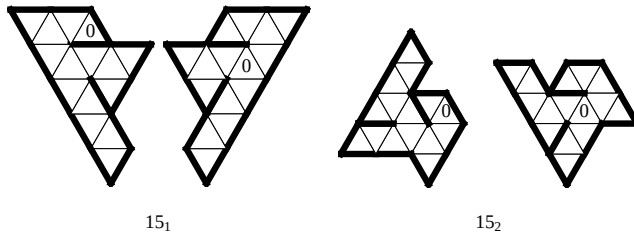


```
Out[103]=
```

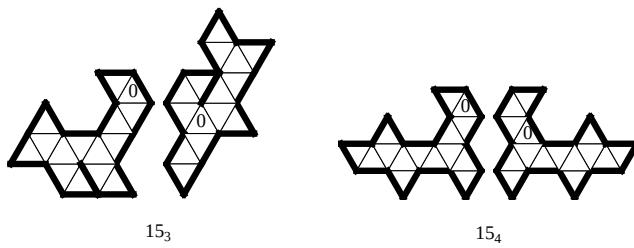


```
In[104]:= "Examples from PSL(4,2).";
```

```
F15 = Show[GraphicsArray[{{F151, F152}, {F153, F154}}]]
```

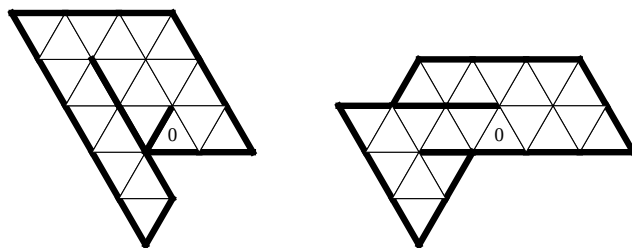


```
Out[105]=
```



```
In[106]:= "Examples from PSL(3,4).";  
F21 = Show[F211]
```

Out[107]=



21_1

```
In[108]:= "Figure special";
```

In[109]:= "Individual diagrams.";

S1 = ShowFig0[D73, 45, 45, 90]

S2 = ShowFig0[D73, 60, 45, 75]

S3 = ShowFig0[D71, 60, 45, 75]

S4 = ShowFig0[D72, 60, 45, 75]

S5 = ShowFig0[D73, 60, 90, 30]

S6 = ShowFig0[D71, 30, 60, 90]

S7 = ShowFig0[D138, 60, 30, 90]

S8 = ShowFig0[D211, 90, 60, 30]

S9 = ShowFig0[D72, 70, 30, 80]

S10 = ShowFig0[D73, 30, 70, 80]

S11 = ShowFig0[D134, 40, 90, 50]

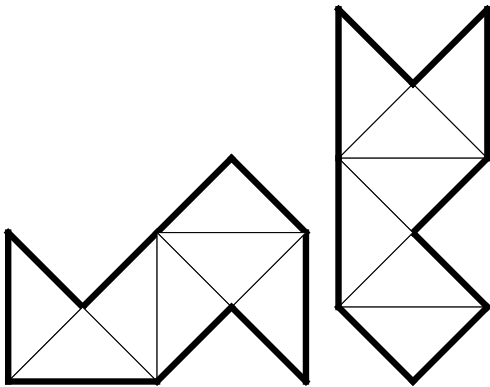
S12 = ShowFig0[D135, 110, 30, 40]

S13 = ShowFig0[D136, 58, 66, 56]

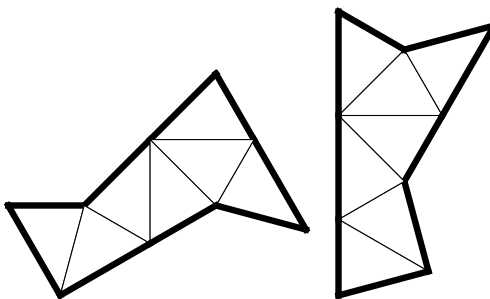
S14 = ShowFig0[D154, 60, 50, 70]

S15 = ShowFig0[D73, 40, 30, 110]

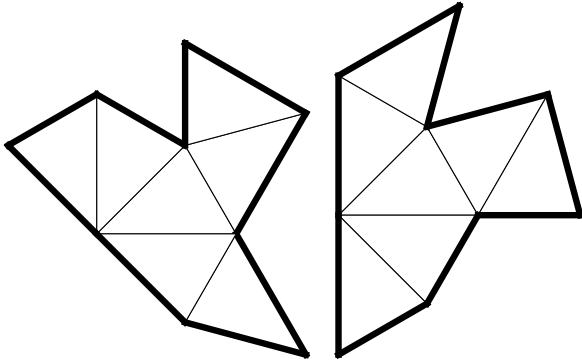
Out[110]=



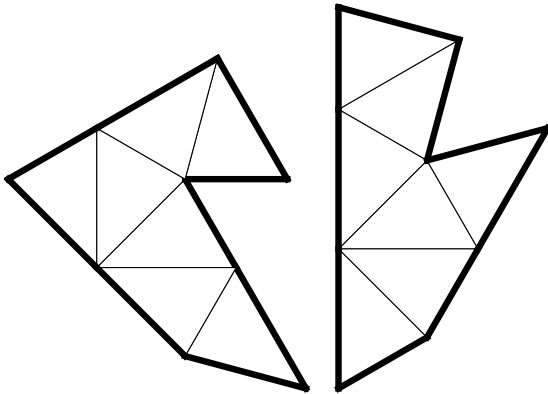
Out[111]=



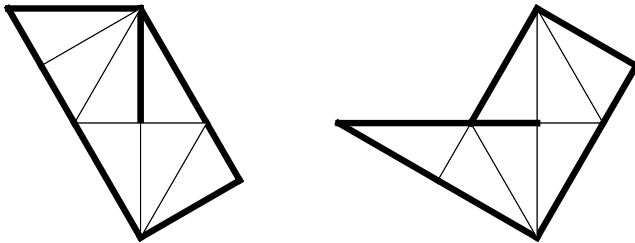
Out[112]=



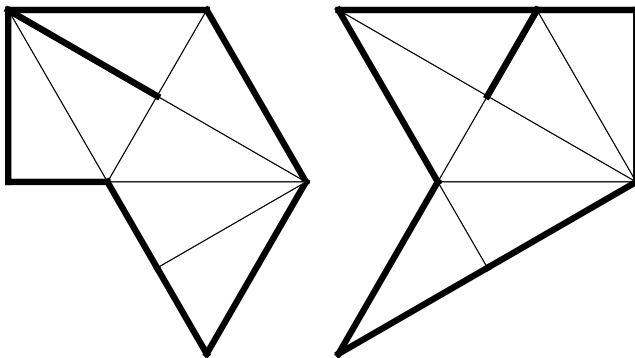
Out[113]=



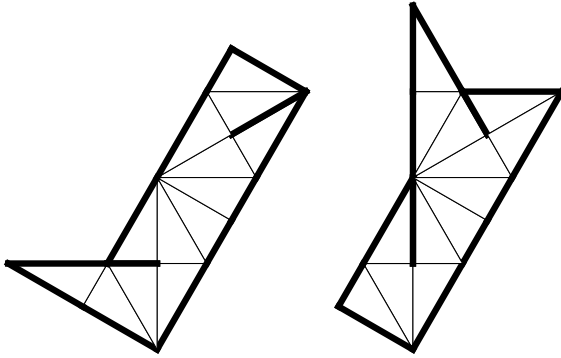
Out[114]=



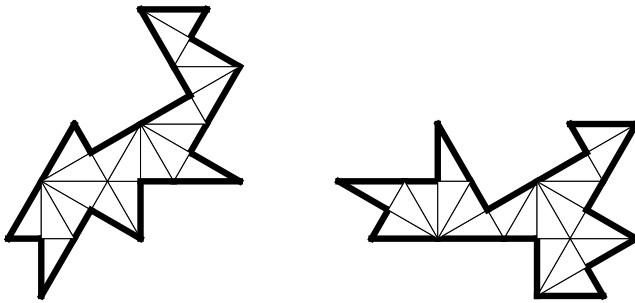
Out[115]=



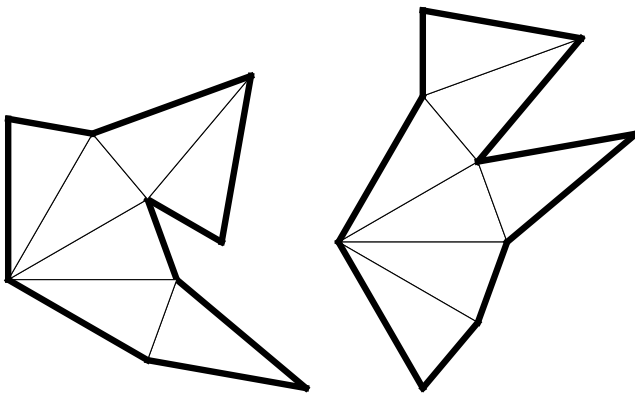
Out[116]=



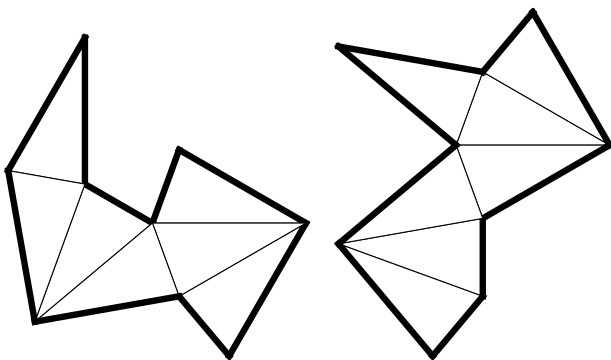
Out[117]=



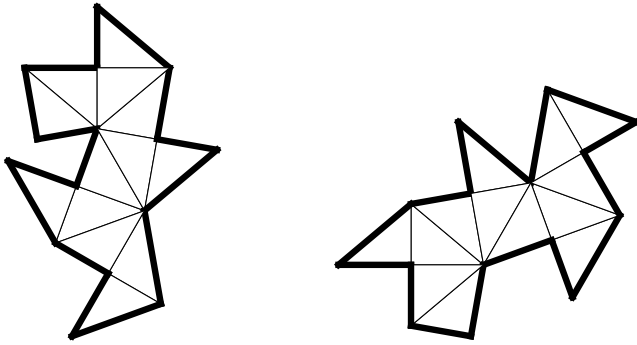
Out[118]=



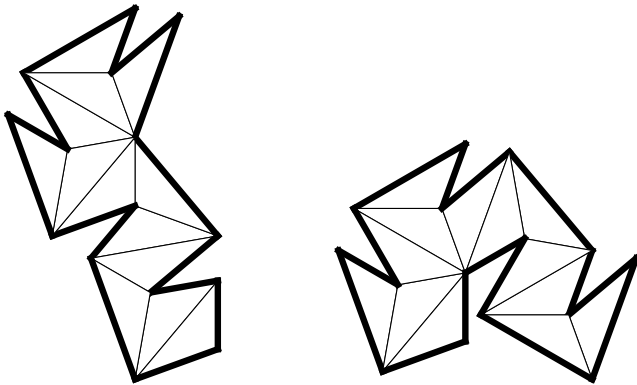
Out[119]=



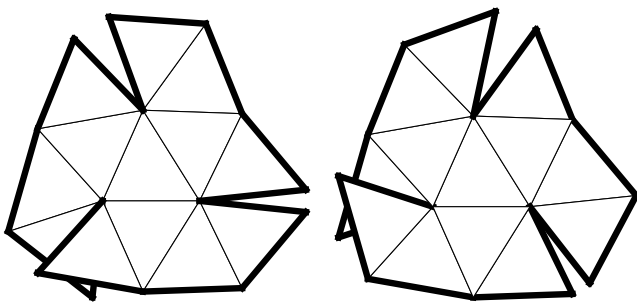
Out[120]=



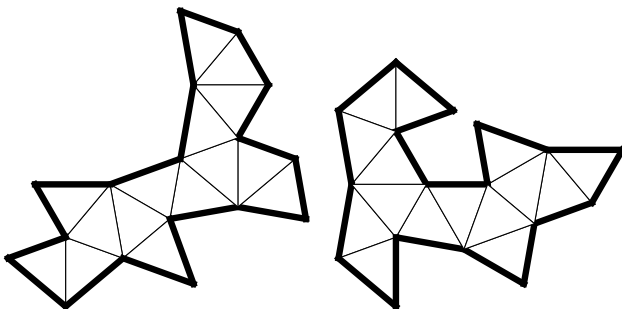
Out[121]=



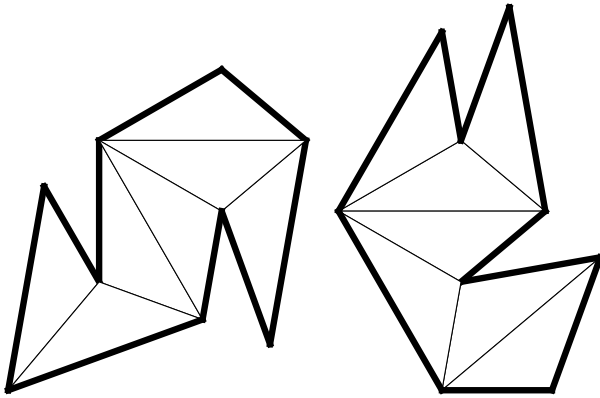
Out[122]=



Out[123]=



Out[124]=



```
In[125]:= "The figure itself.";
```

```
Show[
  GraphicsArray[{{S1, S2, S3}, {S4, S5, S6}, {S7, S8, S9}, {S10, S11, S12}, {S13, S14, S15}}]]
```

