

The Penrose package

Andrew Stacey

loopspace@mathforge.org

1.0 from 2018/07/10

1 Introduction

This is a TikZ library for drawing Penrose tiles (kite/dart, rhombus, and pentagon versions). It provides two methods of drawing: one in which an automatic pattern is built, and one where the tiles can be placed “by hand”. The tiles can be shaped and (hopefully!) still fit together. For user documentation, see the `penrose.pdf` file.

2 Implementation

2.1 Initialisation

```
1 <@=@penrose>
```

We use the `spath3` library for manipulating the paths that will make up the tiles.

```
2 \RequirePackage{spath3}
```

Now we move in to the realm of $\text{\LaTeX}3$.

```
3 \ExplSyntaxOn
```

Start with some basic paths (lines) for the sides of the tiles so that we know that we have well-defined tiles at the outset.

```
4 \MakeSPath{Penrose path a}
```

```
5 {
```

```
6     \pgfsyssoftpath@movetotoken{0pt}{0pt}
```

```
7     \pgfsyssoftpath@linetotoken{1pt}{0pt}
```

```
8 }
```

```
9 \SPathPrepare{Penrose path a}
```

```
10 \CloneSPath {Penrose path a}{Penrose path b}
```

```
11 \CloneSPath {Penrose path a}{Penrose path c}
```

```
12 \CloneSPath {Penrose path a}{Penrose path d}
```

```
13 \CloneSPath {Penrose path a}{Penrose path A}
```

```
14 \CloneSPath {Penrose path a}{Penrose path B}
```

```
15 \CloneSPath {Penrose path a}{Penrose path C}
```

```
16 \CloneSPath {Penrose path a}{Penrose path D}
```

```

\l__penrose_tmpa_fp
\l__penrose_tmpb_fp
\l__penrose_tmpc_fp
\l__penrose_tmpa_tl
\l__penrose_tmpb_tl
\l__penrose_tmpc_tl

```

We need a few temporary variables to hold intermediate calculations.

```

17 \fp_new:N \l__penrose_tmpa_fp
18 \fp_new:N \l__penrose_tmpb_fp
19 \fp_new:N \l__penrose_tmpc_fp
20 \tl_new:N \l__penrose_tmpa_tl
21 \tl_new:N \l__penrose_tmpb_tl
22 \tl_new:N \l__penrose_tmpc_tl

```

2.2 Creating the Tiles

`\penrose_normalise_path:n`

When defining the path for a side, we normalise so that it starts at the origin and ends at (1pt,0pt).

```

23 \cs_new_nopar:Npn \penrose_normalise_path:n #1
24 {

```

Get the initial point of the path and convert to floating point.

```

25 \spath_get:nnN {#1} {initial point} \l__penrose_tmpa_tl
26 \fp_set:Nn \l__penrose_tmpa_fp {\tl_head:N \l__penrose_tmpa_tl}
27 \tl_set:Nx \l__penrose_tmpa_tl {\tl_tail:N \l__penrose_tmpa_tl}
28 \fp_set:Nn \l__penrose_tmpb_fp {\tl_head:N \l__penrose_tmpa_tl}

```

Get the final point of the path, and compute the difference of the final and initial points.

The resulting numbers, say a and b , will be put into a matrix to rotate and scale the path. The formula for the matrix is:

$$\frac{1}{a^2 + b^2} \begin{bmatrix} a & b \\ -b & a \end{bmatrix}$$

```

29 \spath_get:nnN {#1} {final point} \l__penrose_tmpa_tl
30 \fp_set:Nn \l__penrose_tmpa_fp
31 {\tl_head:N \l__penrose_tmpa_tl - \l__penrose_tmpa_fp}
32 \tl_set:Nx \l__penrose_tmpa_tl {\tl_tail:N \l__penrose_tmpa_tl}
33 \fp_set:Nn \l__penrose_tmpb_fp
34 {\tl_head:N \l__penrose_tmpa_tl - \l__penrose_tmpb_fp}

```

Now compute the square of the length of the path for scaling.

```

35 \fp_set:Nn \l__penrose_tmpc_fp
36 {\l__penrose_tmpa_fp^2 + \l__penrose_tmpb_fp^2}
37 \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_tmpa_fp/\l__penrose_tmpc_fp}
38 \fp_set:Nn \l__penrose_tmpb_fp {\l__penrose_tmpb_fp/\l__penrose_tmpc_fp}
39 \fp_set:Nn \l__penrose_tmpc_fp {-\l__penrose_tmpb_fp}

```

Now construct the matrix.

```

40 \tl_set:Nx \l__penrose_tmpb_tl
41 {
42   {\fp_use:N \l__penrose_tmpa_fp}
43   {\fp_use:N \l__penrose_tmpb_fp}
44   {\fp_use:N \l__penrose_tmpc_fp}
45   {\fp_use:N \l__penrose_tmpa_fp}
46 }

```

Get the initial point back again for the translation part.

```

47 \spath_get:nnN {#1} {initial point} \l__penrose_tmpa_tl

```

But we need to premultiply by the matrix because of how the transformations are applied.

```

48 \fp_set:Nn \l__penrose_tmpa_fp
49 {
50   (-1) * \l__penrose_tmpa_fp * \tl_head:N \l__penrose_tmpa_tl
51   + (-1) * \l__penrose_tmpb_fp * \tl_tail:N \l__penrose_tmpa_tl
52 }
53 \fp_set:Nn \l__penrose_tmpb_fp
54 {
55   (-1) * \l__penrose_tmpa_fp * \tl_tail:N \l__penrose_tmpa_tl
56   + \l__penrose_tmpb_fp * \tl_head:N \l__penrose_tmpa_tl
57 }

```

Finally, we apply the transformation to the path.

```

58 \tl_put_right:Nx \l__penrose_tmpb_tl {
59   {\fp_to_dim:N \l__penrose_tmpa_fp}
60   {\fp_to_dim:N \l__penrose_tmpb_fp}
61 }
62 \spath_transform:nV {#1} \l__penrose_tmpb_tl
63 }

```

(End definition for \penrose_normalise_path:n.)

`\SetPenrosePath` This sets the path corresponding to a particular side to the current path, and normalises it.

```

64 \NewDocumentCommand \SetPenrosePath { m }
65 {
66   \pgfsyssoftpath@getcurrentpath\l__penrose_tmpa_tl
67   \spath_clear_new:n {Penrose path #1}
68   \spath_put:nnV {Penrose path #1} {path} \l__penrose_tmpa_tl
69   \penrose_normalise_path:n {Penrose path #1}
70 }

```

(End definition for \SetPenrosePath.)

`\tikz_scan_point:n` This is a wrapper around `\tikz@scan@one@point` to make it easier to use with L^AT_EX₃ variables.

```

71 \cs_new_nopar:Npn \tikz_scan_point:n #1
72 {
73   \tikz@scan@one@point\pgfutil@firstofone#1\relax
74 }
75 \cs_generate_variant:Nn \tikz_scan_point:n {V}

```

(End definition for \tikz_scan_point:n.)

`\penrose_make_tile:nnn` This builds the tile path from its pieces. The arguments are the name of the tile, the descriptions of the sides, and a token list of the coordinates.

```

76 \cs_new_nopar:Npn \penrose_make_tile:nnn #1#2#3
77 {

```

Get the first coordinate and initialise the path with a move to this point.

```

78 \tl_set:Nn \l__penrose_tmpa_tl {#3}
79 \tl_set:Nx \l__penrose_tmpb_tl {\tl_head:N \l__penrose_tmpa_tl}
80 \tl_set:Nn \l__penrose_tmpa_tl {\pgfsyssoftpath@movetotoken}
81 \tikz_scan_point:V \l__penrose_tmpb_tl
82 \tl_put_right:Nx \l__penrose_tmpa_tl

```

```

83 {
84   {\dim_use:N \pgf@x}{\dim_use:N \pgf@y}
85 }
86 \spath_clear_new:n {Penrose path tile #1}
87 \spath_put:nnV {Penrose path tile #1} {path} \l__penrose_tmpa_tl

```

Now we have our path initialised, we can start appending the side paths according to the specification in the second argument.

We append the initial coordinate to the end of the list to make a closed cycle.

```

88 \tl_set:Nn \l__penrose_tmpa_tl {#3}
89 \tl_put_right:Nx \l__penrose_tmpa_tl {{\tl_head:N \l__penrose_tmpa_tl}}

```

Now we walk through the description of the sides, adding the specified paths to our tile path.

```

90 \tl_map_inline:nn {#2} {

```

Clone the path for this side.

```

91   \spath_clone:nn {Penrose path ##1} {Penrose path tmpa}

```

Strip off the next coordinate, and convert it to a point.

```

92   \tl_set:Nx \l__penrose_tmpb_tl {\tl_head:N \l__penrose_tmpa_tl}
93   \tl_set:Nx \l__penrose_tmpa_tl {\tl_tail:N \l__penrose_tmpa_tl}
94   \tikz_scan_point:V \l__penrose_tmpb_tl

```

Store the resulting coordinate.

```

95   \fp_set:Nn \l__penrose_tmpa_fp { \pgf@x }
96   \fp_set:Nn \l__penrose_tmpb_fp { \pgf@y }

```

Now get the next coordinate.

```

97   \tl_set:Nx \l__penrose_tmpb_tl {\tl_head:N \l__penrose_tmpa_tl}
98   \tikz_scan_point:V \l__penrose_tmpb_tl

```

We want the difference between the two coordinates.

```

99   \fp_set:Nn \l__penrose_tmpa_fp {\pgf@x - \l__penrose_tmpa_fp}
100  \fp_set:Nn \l__penrose_tmpb_fp {\pgf@y - \l__penrose_tmpb_fp}

```

This is converted into a transformation matrix.

```

101  \fp_set:Nn \l__penrose_tmpc_fp {-\l__penrose_tmpb_fp}
102  \tl_set:Nx \l__penrose_tmpb_tl
103  {
104    {\fp_use:N \l__penrose_tmpa_fp}
105    {\fp_use:N \l__penrose_tmpc_fp}
106    {\fp_use:N \l__penrose_tmpb_fp}
107    {\fp_use:N \l__penrose_tmpa_fp}
108    {0}
109    {0}
110  }

```

The transformation is applied to the cloned path.

```

111  \spath_transform:nV {Penrose path tmpa} \l__penrose_tmpb_tl

```

And this is welded to the tile path.

```

112  \spath_weld:nn {Penrose path tile #1} {Penrose path tmpa}
113  }

```

At the end we close the path.

```

114  \spath_close_path:n {Penrose path tile #1}
115  }

```

(End definition for `\penrose_make_tile:nnn`.)

`\penrose_make_tile:nn` A wrapper around the above which allows us to specify the second two arguments as two items in a token list.

```

116 \cs_new_nopar:Npn \penrose_make_tile:nn #1#2
117 {
118   \penrose_make_tile:nnn {#1} #2
119 }
120 \cs_generate_variant:Nn \penrose_make_tile:nn {nV}

```

(End definition for `\penrose_make_tile:nn`.)

2.3 Specifying the Tiles

The tile specifications are contained in a `prop`.

```

121 \prop_new:N \g__penrose_tiles_prop

```

`\tl_add_coordinate:Nnn` Process a coordinate through `fp` and adds it to a token list.

```

122 \cs_new_nopar:Npn \tl_add_coordinate:Nnn #1#2#3 {
123   \fp_set:Nn \l__penrose_tmpa_fp{#2}
124   \fp_set:Nn \l__penrose_tmpb_fp{#3}
125   \tl_put_right:Nx #1
126   {
127     {(\fp_use:N \l__penrose_tmpa_fp, \fp_use:N \l__penrose_tmpb_fp)}
128   }
129 }

```

(End definition for `\tl_add_coordinate:Nnn`.)

Now we specify the tiles. The specification is a clockwise list of the vertices together with the labels of the corresponding sides. There are three basic paths, `a`, `b`, `c`, and their complements (which are capitalised).

- Thin Rhombus.

```

130   \tl_clear:N \l__penrose_tmpa_tl
131   \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {0}{0}
132   \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {cosd(18)}{sind(18)}
133   \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {2*cosd(18)}{0}
134   \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {cosd(18)}{-sind(18)}
135
136   \prop_gput:Nnx \g__penrose_tiles_prop {thin~ rhombus}
137   {{a A B b} {\tl_use:N \l__penrose_tmpa_tl}}

```

- Thick Rhombus.

```

138   \tl_clear:N \l__penrose_tmpa_tl
139   \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {0}{0}
140   \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {cosd(36)}{sind(36)}
141   \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {2*cosd(36)}{0}
142   \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {cosd(36)}{-sind(36)}
143
144   \prop_gput:Nnx \g__penrose_tiles_prop {thick~ rhombus}
145   {{B a A b} {\tl_use:N \l__penrose_tmpa_tl}}

```

- Dart.

```

146 \tl_clear:N \l__penrose_tmpa_tl
147 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {0}{0}
148 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl
149 {2*sind(18)*cosd(108)}{2*sind(18)*sind(108)}
150 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {2*sind(18)}{0}
151 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl
152 {2*sind(18)*cosd(108)}{-2*sind(18)*sind(108)}
153
154 \prop_gput:Nnx \g__penrose_tiles_prop {dart}
155 {{c a A C} {\tl_use:N \l__penrose_tmpa_tl}}

```

- Kite.

```

156 \tl_clear:N \l__penrose_tmpa_tl
157 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {0}{0}
158 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {cosd(36)}{sind(36)}
159 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {1}{0}
160 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {cosd(36)}{-sind(36)}
161
162 \prop_gput:Nnx \g__penrose_tiles_prop {kite}
163 {{a c C A} {\tl_use:N \l__penrose_tmpa_tl}}

```

- Golden Triangle.

```

164 \tl_clear:N \l__penrose_tmpa_tl
165 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {0}{0}
166 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {cosd(18)}{sind(18)}
167 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {cosd(18)}{-sind(18)}
168
169 \prop_gput:Nnx \g__penrose_tiles_prop {golden~ triangle}
170 {{a c b} {\tl_use:N \l__penrose_tmpa_tl}}

```

- Reverse Golden Triangle.

```

171 \tl_clear:N \l__penrose_tmpa_tl
172 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {0}{0}
173 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {cosd(18)}{sind(18)}
174 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {cosd(18)}{-sind(18)}
175
176 \prop_gput:Nnx \g__penrose_tiles_prop {reverse~ golden~ triangle}
177 {{B C A} {\tl_use:N \l__penrose_tmpa_tl}}

```

- Golden Gnomon

```

178 \tl_clear:N \l__penrose_tmpa_tl
179 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {0}{0}
180 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {cosd(36)}{sind(36)}
181 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {2*cosd(36)}{0}
182
183 \prop_gput:Nnx \g__penrose_tiles_prop {golden~ gnomon}
184 {{C b A} {\tl_use:N \l__penrose_tmpa_tl}}

```

- Reverse Golden Gnomon

```

185 \tl_clear:N \l__penrose_tmpa_tl
186 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {0}{0}
187 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {2*cosd(36)}{0}
188 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {cosd(36)}{-sind(36)}
189 \prop_gput:Nnx \g__penrose_tiles_prop {reverse~ golden~ gnomon}
190 {{a B c} {\tl_use:N \l__penrose_tmpa_tl}}

```

- Primary Pentagon (pentagon 5)

```

191 \tl_clear:N \l__penrose_tmpa_tl
192 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {0}{0}
193 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {cosd(108)}{sind(108)}
194 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl
195 {1+cosd(72)+cosd(144)}{sind(72)+sind(144)}
196 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {1+cosd(72)}{sind(72)}
197 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {1}{0}
198 \prop_gput:Nnx \g__penrose_tiles_prop {pentagon~ 5}
199 {{a a a a a} {\tl_use:N \l__penrose_tmpa_tl}}

```

- Secondary Pentagon (pentagon 3)

```

200 \prop_gput:Nnx \g__penrose_tiles_prop {pentagon~ 3}
201 {{A b a a b} {\tl_use:N \l__penrose_tmpa_tl}}

```

- Tertiary Pentagon (pentagon 2)

```

202 \prop_gput:Nnx \g__penrose_tiles_prop {pentagon~ 2}
203 {{d A c c A} {\tl_use:N \l__penrose_tmpa_tl}}

```

- Pentagram

```

204 \tl_clear:N \l__penrose_tmpa_tl
205 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {1}{0}
206 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {1-cosd(36)}{-sind(36)}
207 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl
208 {1-cosd(36)-cosd(108)}{-sind(36)-sind(108)}
209 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {cosd(108)}{-sind(108)}
210 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl
211 {-1+3*cosd(108)+cosd(36)}{-sind(36)-sind(108)}
212 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl
213 {-1+2*cosd(108)+cosd(36)}{-sind(36)}
214 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {-1+2*cosd(108)}{0}
215 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {2*cosd(108)}{0}
216 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {cosd(108)}{sind(108)}
217 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {0}{0}
218 \prop_gput:Nnx \g__penrose_tiles_prop {pentagram}
219 {{C C C C C C C C} {\tl_use:N \l__penrose_tmpa_tl}}

```

- Boat

```

220 \tl_clear:N \l__penrose_tmpa_tl

```

```

221 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {-1+2*cosd(108)}{0}
222 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {2*cosd(108)}{0}
223 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {cosd(108)}{sind(108)}
224 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {0}{0}
225 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {1}{0}
226 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {1-cosd(36)}{-sind(36)}
227 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl
228 {-1+2*cosd(108)+cosd(36)}{-sind(36)}
229 \prop_gput:Nnx \g__penrose_tiles_prop {boat}
230 {{C C C C B D B} {\tl_use:N \l__penrose_tmpa_tl}}

```

- Diamond.

```

231 \tl_clear:N \l__penrose_tmpa_tl
232 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {0}{0}
233 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {cosd(18)}{sind(18)}
234 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {2*cosd(18)}{0}
235 \tl_add_coordinate:Nnn \l__penrose_tmpa_tl {cosd(18)}{-sind(18)}
236 \prop_gput:Nnx \g__penrose_tiles_prop {diamond}
237 {{D B B D} {\tl_use:N \l__penrose_tmpa_tl}}

```

`\MakePenroseTile` This is the user wrapper around the tile creation macros.

```

238 \NewDocumentCommand \MakePenroseTile {m}
239 {
240 \prop_get:NnN \g__penrose_tiles_prop {#1} \l__penrose_tmpa_tl
241 \penrose_make_tile:nV {#1} \l__penrose_tmpa_tl
242 }

```

(End definition for \MakePenroseTile.)

`\UsePenroseTile` This is the command that actually places a tile on the page. The first argument is optional and is for styling.

```

243 \NewDocumentCommand \UsePenroseTile {0{ } m}
244 {

```

We need to transform the tile to correspond to the current transformation matrix. To ensure that we only transform the current tile, we clone it first.

```

245 \spath_clone:nn {Penrose path tile #2} {Penrose path tmpa}

```

The transformation matrix returned by PGF appears to be transposed from what it should be. (This needs a little more investigation, it might be that I've implemented the multiplication incorrectly here.)

```

246 \pgfgettransform \l__penrose_tmpa_tl
247 \tl_clear:N \l__penrose_tmpb_tl
248 \tl_set:Nx \l__penrose_tmpb_tl {{\tl_head:N \l__penrose_tmpa_tl}}
249 \tl_set:Nx \l__penrose_tmpa_tl {\tl_tail:N \l__penrose_tmpa_tl}
250 \tl_put_right:Nx \l__penrose_tmpb_tl
251 {
252 {\tl_item:Nn \l__penrose_tmpa_tl {2}}
253 }
254 \tl_put_right:Nx \l__penrose_tmpb_tl
255 {
256 {\tl_item:Nn \l__penrose_tmpa_tl {1}}
257 }

```



```

258 \tl_set:Nx \l__penrose_tmpa_tl {\tl_tail:N \l__penrose_tmpa_tl}
259 \tl_set:Nx \l__penrose_tmpa_tl {\tl_tail:N \l__penrose_tmpa_tl}
260 \tl_put_right:NV \l__penrose_tmpb_tl \l__penrose_tmpa_tl

```

Apply the transformation, protocol the path, and render it.

```

261 \spath_transform:nV {Penrose path tmpa} \l__penrose_tmpb_tl
262 \spath_protocol_path:n {Penrose path tmpa}
263 \spath_tikz_path:nn {#1}{Penrose path tmpa}
264 }

```

(End definition for \UsePenroseTile.)

This is a style for a user to take a path and make it into the path for one of the sides. It needs to store both that side and the reverse.

```

265 \tikzset{
266   save~ Penrose~ path/.code={
267     \tikz@addmode{

```

Get the current path.

```

268       \pgfsyssoftpath@getcurrentpath\l__penrose_tmpa_tl

```

Clear the receiving path, and store the current path in it.

```

269       \spath_clear_new:n {Penrose path #1}
270       \spath_put:nnV {Penrose path #1} {path} \l__penrose_tmpa_tl

```

Normalise the path.

```

271       \penrose_normalise_path:n {Penrose path #1}

```

Now create the reverse path. The name is the upper case version.

```

272       \tex_uppercase:D {\tl_set:Nx \l__penrose_tmpa_tl {#1}}

```

Clone the path.

```

273       \spath_clone:nn {Penrose path #1}
274       {Penrose path \tl_use:N \l__penrose_tmpa_tl }

```

Reverse it.

```

275       \spath_reverse:n {Penrose path \tl_use:N \l__penrose_tmpa_tl}

```

Swap the start and end.

```

276       \spath_transform:nnnnnn {Penrose path \tl_use:N \l__penrose_tmpa_tl}
277       {-1} {0} {0} {-1} {1} {0}
278     }
279   },
280   expand~ key/.code={
281     \exp_args:NV \pgfkeysalso #1
282   }
283 }

```

Create the basic tile shapes.

```

284 \MakePenroseTile {thin~ rhombus}
285 \MakePenroseTile {thick~ rhombus}
286 \MakePenroseTile {dart}
287 \MakePenroseTile {kite}
288 \MakePenroseTile {golden~ triangle}
289 \MakePenroseTile {reverse~ golden~ triangle}
290 \MakePenroseTile {golden~ gnomon}
291 \MakePenroseTile {reverse~ golden~ gnomon}
292 \MakePenroseTile {pentagon~ 5}

```

```

293 \MakePenroseTile {pentagon~ 3}
294 \MakePenroseTile {pentagon~ 2}
295 \MakePenroseTile {pentagram}
296 \MakePenroseTile {boat}
297 \MakePenroseTile {diamond}

```

2.4 Lindenmayer System

This is an implementation of the Lindenmayer System description of Penrose tilings as a way of generating tilings from a specific starting seed.

The implementation uses `props` to store *rules* and *actions*. The rules are used to expand the starting seed to a certain level, after which the actions are carried out. The syntax is based on the PGF library, but as we're already using L^AT_EX3 it is reimplemented in that.

These are the rules for generating rhombus tilings.

```

298 \prop_new:N \g__penrose_rhombus_lms_rule_prop
299 \prop_put:Nnn \g__penrose_rhombus_lms_rule_prop {T} {[f*sT] [f>g]}
300 \prop_put:Nnn \g__penrose_rhombus_lms_rule_prop {t} {[f_st] [f>G]}
301 \prop_put:Nnn \g__penrose_rhombus_lms_rule_prop {G} {[f+sG] [sf>g] [sf*sT]}
302 \prop_put:Nnn \g__penrose_rhombus_lms_rule_prop {g} {[f-sg] [sf>G] [sf_st]}

```

These are the rules for generating kite and dart tilings.

```

303 \prop_new:N \g__penrose_kite_lms_rule_prop
304 \prop_put:Nnn \g__penrose_kite_lms_rule_prop {T} {[f*sT] [f>st] [+sg]}
305 \prop_put:Nnn \g__penrose_kite_lms_rule_prop {t} {[f_st] [f>st] [-sG]}
306 \prop_put:Nnn \g__penrose_kite_lms_rule_prop {G} {[f**sG] [sT]}
307 \prop_put:Nnn \g__penrose_kite_lms_rule_prop {g} {[f-sg] [st]}

```

These are the rules for generating pentagon tilings.

```

308 \prop_new:N \g__penrose_pentagon_lms_rule_prop
309 \prop_put:Nnn \g__penrose_pentagon_lms_rule_prop {P}
310 {[s>P] [1sF+Q] [1+sF+Q] [1*sF+Q] [1-sF+Q] [1_sF+Q]} % pentagon 5
311 \prop_put:Nnn \g__penrose_pentagon_lms_rule_prop {Q}
312 {[s>P] [1+sFR] [1*sF*R] [1-sF+Q] [1_sF+Q] [1sF+Q] [->fsD]} % pentagon 3
313 \prop_put:Nnn \g__penrose_pentagon_lms_rule_prop {R}
314 {[s>P] [1-sF+Q] [1+sF*R] [1*sFR] [1_sF*R] [1sFR] [->fsD] [>fsD]} % pentagon 2
315 \prop_put:Nnn \g__penrose_pentagon_lms_rule_prop {G}
316 {
317   [s>G]
318   [se [>d+R] [e1B]]
319   [+se [>d+R] [e1B]]
320   [-se [>d+R] [e1B]]
321   [*se [>d+R] [e1B]]
322   [_se [>d+R] [e1B]]
323 } % pentagram
324 \prop_put:Nnn \g__penrose_pentagon_lms_rule_prop {B}
325 {
326   [s>G]
327   [se [>d+R] [e1B]]
328   [+se [>d+R] [e1B]]
329   [-se [>d+R] [e1B]]
330 } % boat
331 \prop_put:Nnn \g__penrose_pentagon_lms_rule_prop {D}
332 {[s>d+R] [s>eG] [se1B]} % diamond

```

Each of the standard tilings can also be drawn using triangles using the same rules.

```

333 \prop_set_eq:NN \g__penrose_rtriangle_lms_rule_prop
334 \g__penrose_rhombus_lms_rule_prop
335 \prop_set_eq:NN \g__penrose_ktriangle_lms_rule_prop
336 \g__penrose_kite_lms_rule_prop

```

These hold the various actions.

```

337 \prop_new:N \g__penrose_default_lms_action_prop
338 \prop_new:N \g__penrose_rhombus_lms_action_prop
339 \prop_new:N \g__penrose_kite_lms_action_prop
340 \prop_new:N \g__penrose_rtriangle_lms_action_prop
341 \prop_new:N \g__penrose_ktriangle_lms_action_prop
342 \prop_new:N \g__penrose_pentagon_lms_action_prop

```

We need some parameters.

```

343 \dim_new:N \l__penrose_step_dim
344 \dim_set:Nn \l__penrose_step_dim {1cm}

```

These are the defaults, which will be used in all the rule sets.

```

345 \prop_put:Nnn \g__penrose_default_lms_action_prop {[]} {\group_begin:}
346 \prop_put:Nnn \g__penrose_default_lms_action_prop {} {\group_end:}
347 \prop_put:Nnn \g__penrose_default_lms_action_prop {f}
348 {\pgftransformxshift{\l__penrose_step_dim}}
349 \prop_put:Nnn \g__penrose_default_lms_action_prop {s} {
350   \fp_set:Nn \l__penrose_tmpa_fp { 2 * sind(18) * \l__penrose_step_dim }
351   \dim_set:Nn \l__penrose_step_dim {\fp_to_dim:N \l__penrose_tmpa_fp}
352 }

```

The rhombus rules need a variety of turns.

```

353 \prop_put:Nnn \g__penrose_rhombus_lms_action_prop {+}
354 {\pgftransformrotate{144}}
355 \prop_put:Nnn \g__penrose_rhombus_lms_action_prop {*}
356 {\pgftransformrotate{108}}
357 \prop_put:Nnn \g__penrose_rhombus_lms_action_prop {-}
358 {\pgftransformrotate{216}}
359 \prop_put:Nnn \g__penrose_rhombus_lms_action_prop {_}
360 {\pgftransformrotate{252}}
361 \prop_put:Nnn \g__penrose_rhombus_lms_action_prop {>}
362 {\pgftransformrotate{180}}

```

Up to now, the actions for the rhombus and its triangle replacement are the same.

```

363 \prop_set_eq:NN \g__penrose_rtriangle_lms_action_prop
364 \g__penrose_rhombus_lms_action_prop

```

Now we do the actions that actually draw something.

```

365 \prop_put:Nnn \g__penrose_rhombus_lms_action_prop {T} {
366   \group_begin:

```

As we go through, we keep track of how many tiles we've drawn.

```

367   \int_gincr:N \l__penrose_tile_int

```

Set up the position, size, and angle correctly.

```

368   \pgftransformrotate{198}
369   \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim*2*cosd(18)}
370   \pgftransformxshift{-\fp_to_dim:N \l__penrose_tmpa_fp}
371   \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim/(1cm)}
372   \pgftransformscale{\fp_use:N \l__penrose_tmpa_fp}

```

Now we draw the thin rhombus, applying every style we can possibly imagine. The `Penrose tile` style gets the current tile and total tile numbers passed to it.

```

373 \tl_set:Nx \l__penrose_tmpc_tl
374 {
375   {\int_use:N \l__penrose_tile_int}
376   {\int_use:N \l__penrose_tiles_int}
377 }
378 \UsePenroseTile[
379   every~ Penrose~ tile/.try,
380   every~ thin~ rhombus/.try,
381   Penrose~ tile~ \int_use:N \l__penrose_tile_int/.try,
382   Penrose~ tile/.try/.expand~ once=\l__penrose_tmpc_tl
383 ]{thin~rhombus}
384 \group_end:
385 }

```

Same for the thick rhombus.

```

386 \prop_put:Nnn \g__penrose_rhombus_lms_action_prop {G} {
387   \group_begin:
388   \int_gincr:N \l__penrose_tile_int
389   \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim/(1cm)/(2*cosd(36))}
390   \pgftransformscalet{\fp_use:N \l__penrose_tmpa_fp}
391   \tl_set:Nx \l__penrose_tmpc_tl
392   {
393     {\int_use:N \l__penrose_tile_int}
394     {\int_use:N \l__penrose_tiles_int}
395   }
396   \UsePenroseTile[
397     every~ Penrose~ tile/.try,
398     every~ thick~ rhombus/.try,
399     Penrose~ tile~ \int_use:N \l__penrose_tile_int/.try,
400     Penrose~ tile/.try/.expand~ once=\l__penrose_tmpc_tl
401   ]{thick~rhombus}
402   \group_end:
403 }

```

Now we do the same for the kite and dart tiling.

```

404 \prop_put:Nnn \g__penrose_kite_lms_action_prop {+} {\pgftransformrotate{36}}
405 \prop_put:Nnn \g__penrose_kite_lms_action_prop {*} {\pgftransformrotate{108}}
406 \prop_put:Nnn \g__penrose_kite_lms_action_prop {-} {\pgftransformrotate{-36}}
407 \prop_put:Nnn \g__penrose_kite_lms_action_prop {_}
408 {\pgftransformrotate{-108}}
409 \prop_put:Nnn \g__penrose_kite_lms_action_prop {>} {\pgftransformrotate{180}}
410 \prop_set_eq:NN \g__penrose_ktriangle_lms_action_prop
411 \g__penrose_kite_lms_action_prop
412 \prop_put:Nnn \g__penrose_kite_lms_action_prop {T} {
413   \group_begin:
414   \int_gincr:N \l__penrose_tile_int
415   \pgftransformrotate{36}
416   \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim/(1cm)}
417   \pgftransformscalet{\fp_use:N \l__penrose_tmpa_fp}
418   \tl_set:Nx \l__penrose_tmpc_tl
419   {
420     {\int_use:N \l__penrose_tile_int}

```

```

421   {\int_use:N \l__penrose_tiles_int}
422   }
423   \UsePenroseTile[
424     every~ Penrose~ tile/.try,
425     every~ kite/.try,
426     Penrose~ tile~ \int_use:N \l__penrose_tile_int/.try,
427     Penrose~ tile/.try/.expand~ once=\l__penrose_tmpc_tl
428   ]{kite}
429   \group_end:
430 }

431 \prop_put:Nnn \g__penrose_kite_lms_action_prop {g} {
432   \group_begin:
433   \int_gincr:N \l__penrose_tile_int
434   \pgftransformrotate{144}
435   \pgftransformxshift{-\l__penrose_step_dim * 2 * sin(18)}
436   \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim/(1cm)}
437   \pgftransformscale{\fp_use:N \l__penrose_tmpa_fp}
438   \tl_set:Nx \l__penrose_tmpc_tl
439   {
440     {\int_use:N \l__penrose_tile_int}
441     {\int_use:N \l__penrose_tiles_int}
442   }
443   \UsePenroseTile[
444     every~ Penrose~ tile/.try,
445     every~ dart/.try,
446     Penrose~ tile~ \int_use:N \l__penrose_tile_int/.try,
447     Penrose~ tile/.try/.expand~ once=\l__penrose_tmpc_tl
448   ]{dart}
449   \group_end:
450 }

Now we set up the actions for the triangle variations.

451 \prop_put:Nnn \g__penrose_rtriangle_lms_action_prop {T} {
452   \group_begin:
453   \int_gincr:N \l__penrose_tile_int
454   \pgftransformrotate{18}
455   \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim/(1cm)}
456   \pgftransformscale{\fp_use:N \l__penrose_tmpa_fp}
457   \tl_set:Nx \l__penrose_tmpc_tl
458   {
459     {\int_use:N \l__penrose_tile_int}
460     {\int_use:N \l__penrose_tiles_int}
461   }
462   \UsePenroseTile[
463     every~ Penrose~ tile/.try,
464     every~ reverse~ golden~ triangle/.try,
465     Penrose~ tile~ \int_use:N \l__penrose_tile_int/.try,
466     Penrose~ tile/.try/.expand~ once=\l__penrose_tmpc_tl
467   ]{reverse~ golden~ triangle}
468   \group_end:
469 }

470 \prop_put:Nnn \g__penrose_rtriangle_lms_action_prop {t} {
471   \group_begin:
472   \int_gincr:N \l__penrose_tile_int

```

```

473 \pgftransformrotate{-18}
474 \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim/(1cm)}
475 \pgftransformscale{\fp_use:N \l__penrose_tmpa_fp}
476 \tl_set:Nx \l__penrose_tmpc_tl
477 {
478   {\int_use:N \l__penrose_tile_int}
479   {\int_use:N \l__penrose_tiles_int}
480 }
481 \tl_set:Nx \l__penrose_tmpc_tl
482 {
483   {\int_use:N \l__penrose_tile_int}
484   {\int_use:N \l__penrose_tiles_int}
485 }
486 \UsePenroseTile[
487   every~ Penrose~ tile/.try,
488   every~ golden~ triangle/.try,
489   Penrose~ tile~ \int_use:N \l__penrose_tile_int/.try,
490   Penrose~ tile/.try/.expand~ once=\l__penrose_tmpc_tl
491 ]{golden~ triangle}
492 \group_end:
493 }

494 \prop_put:Nnn \g__penrose_rtriangle_lms_action_prop {G} {
495   \group_begin:
496   \int_gincr:N \l__penrose_tile_int
497   \pgftransformrotate{180}
498   \pgftransformxshift{-\l__penrose_step_dim}
499   \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim/(1cm)/(2*cosd(36))}
500   \pgftransformscale{\fp_use:N \l__penrose_tmpa_fp}
501   \tl_set:Nx \l__penrose_tmpc_tl
502   {
503     {\int_use:N \l__penrose_tile_int}
504     {\int_use:N \l__penrose_tiles_int}
505   }
506   \UsePenroseTile[
507     every~ Penrose~ tile/.try,
508     every~ reverse~ golden~ gnomon/.try,
509     Penrose~ tile~ \int_use:N \l__penrose_tile_int/.try,
510     Penrose~ tile/.try/.expand~ once=\l__penrose_tmpc_tl
511   ]{reverse~ golden~ gnomon}
512   \group_end:
513 }

514 \prop_put:Nnn \g__penrose_rtriangle_lms_action_prop {g} {
515   \group_begin:
516   \int_gincr:N \l__penrose_tile_int
517   \pgftransformrotate{180}
518   \pgftransformxshift{-\l__penrose_step_dim}
519   \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim/(1cm)/(2*cosd(36))}
520   \pgftransformscale{\fp_use:N \l__penrose_tmpa_fp}
521   \tl_set:Nx \l__penrose_tmpc_tl
522   {
523     {\int_use:N \l__penrose_tile_int}
524     {\int_use:N \l__penrose_tiles_int}
525   }

```

```

526 \UsePenroseTile[
527   every~ Penrose~ tile/.try,
528   every~ golden~ gnomon/.try,
529   Penrose~ tile~ \int_use:N \l__penrose_tile_int/.try,
530   Penrose~ tile/.try/.expand~ once=\l__penrose_tmpc_tl
531 ]{golden~ gnomon}
532 \group_end:
533 }

534 \prop_put:Nnn \g__penrose_ktriangle_lms_action_prop {T} {
535   \group_begin:
536   \int_gincr:N \l__penrose_tile_int
537   \pgftransformrotate{18}
538   \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim/(1cm)}
539   \pgftransformscale{\fp_use:N \l__penrose_tmpa_fp}
540   \tl_set:Nx \l__penrose_tmpc_tl
541   {
542     {\int_use:N \l__penrose_tile_int}
543     {\int_use:N \l__penrose_tiles_int}
544   }
545   \UsePenroseTile[
546     every~ Penrose~ tile/.try,
547     every~ reverse~ golden~ triangle/.try,
548     Penrose~ tile~ \int_use:N \l__penrose_tile_int/.try,
549     Penrose~ tile/.try/.expand~ once=\l__penrose_tmpc_tl
550   ]{reverse~ golden~ triangle}
551   \group_end:
552 }

553 \prop_put:Nnn \g__penrose_ktriangle_lms_action_prop {t} {
554   \group_begin:
555   \int_gincr:N \l__penrose_tile_int
556   \pgftransformrotate{-18}
557   \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim/(1cm)}
558   \pgftransformscale{\fp_use:N \l__penrose_tmpa_fp}
559   \tl_set:Nx \l__penrose_tmpc_tl
560   {
561     {\int_use:N \l__penrose_tile_int}
562     {\int_use:N \l__penrose_tiles_int}
563   }
564   \UsePenroseTile[
565     every~ Penrose~ tile/.try,
566     every~ golden~ triangle/.try,
567     Penrose~ tile~ \int_use:N \l__penrose_tile_int/.try,
568     Penrose~ tile/.try/.expand~ once=\l__penrose_tmpc_tl
569   ]{golden~ triangle}
570   \group_end:
571 }

572 \prop_put:Nnn \g__penrose_ktriangle_lms_action_prop {G} {
573   \group_begin:
574   \int_gincr:N \l__penrose_tile_int
575   \pgftransformrotate{180}
576   \pgftransformxshift{-\l__penrose_step_dim}
577   \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim/(1cm)/(2*cosd(36))}
578   \pgftransformscale{\fp_use:N \l__penrose_tmpa_fp}

```

```

579 \tl_set:Nx \l__penrose_tmpc_tl
580 {
581   {\int_use:N \l__penrose_tile_int}
582   {\int_use:N \l__penrose_tiles_int}
583 }
584 \UsePenroseTile[
585   every~ Penrose~ tile/.try,
586   every~ reverse~ golden~ gnomon/.try,
587   Penrose~ tile~ \int_use:N \l__penrose_tile_int/.try,
588   Penrose~ tile/.try/.expand~ once=\l__penrose_tmpc_tl
589 ]{reverse~ golden~ gnomon}
590 \group_end:
591 }

592 \prop_put:Nnn \g__penrose_ktriangle_lms_action_prop {g} {
593   \group_begin:
594   \int_gincr:N \l__penrose_tile_int
595   \pgftransformrotate{180}
596   \pgftransformxshift{-\l__penrose_step_dim}
597   \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim/(1cm)/(2*cosd(36))}
598   \pgftransformscale{\fp_use:N \l__penrose_tmpa_fp}
599   \tl_set:Nx \l__penrose_tmpc_tl
600   {
601     {\int_use:N \l__penrose_tile_int}
602     {\int_use:N \l__penrose_tiles_int}
603   }
604   \UsePenroseTile[
605     every~ Penrose~ tile/.try,
606     every~ golden~ gnomon/.try,
607     Penrose~ tile~ \int_use:N \l__penrose_tile_int/.try,
608     Penrose~ tile/.try/.expand~ once=\l__penrose_tmpc_tl
609   ]{golden~ gnomon}
610   \group_end:
611 }

```

Now we do the same for the pentagonal tilings.

The rules need a variety of turns.

```

612 \int_new:N \l__penrose_pentagon_parity_int
613 \seq_new:N \l__penrose_pentagon_parity_seq
614 \seq_set_from_clist:Nn \l__penrose_pentagon_parity_seq {odd,even}
615 \prop_put:Nnn \g__penrose_pentagon_lms_action_prop {1} {
616   \int_set:Nn \l__penrose_pentagon_parity_int
617   {3 - \l__penrose_pentagon_parity_int}
618 }
619 \prop_put:Nnn \g__penrose_pentagon_lms_action_prop {+}
620 {\pgftransformrotate{72}}
621 \prop_put:Nnn \g__penrose_pentagon_lms_action_prop {*}
622 {\pgftransformrotate{144}}
623 \prop_put:Nnn \g__penrose_pentagon_lms_action_prop {-}
624 {\pgftransformrotate{288}}
625 \prop_put:Nnn \g__penrose_pentagon_lms_action_prop {_}
626 {\pgftransformrotate{216}}
627 \prop_put:Nnn \g__penrose_pentagon_lms_action_prop {>}
628 {\pgftransformrotate{180}}
629 \prop_put:Nnn \g__penrose_pentagon_lms_action_prop {}

```


630 {\pgftransformxscale{-1}}

The scale factor is different.

```

631 \prop_put:Nnn \g__penrose_pentagon_lms_action_prop {s} {
632   \fp_set:Nn \l__penrose_tmpa_fp
633   {
634     1/(2 + 2 * cosd(72) ) * \l__penrose_step_dim
635   }
636   \dim_set:Nn \l__penrose_step_dim {\fp_to_dim:N \l__penrose_tmpa_fp}
637 }

```

And we tend to work better vertically.

```

638 \prop_put:Nnn \g__penrose_pentagon_lms_action_prop {f} {
639   \fp_set:Nn \l__penrose_tmpa_fp { tand(54)/2 * \l__penrose_step_dim }
640   \pgftransformyshift{\fp_to_dim:N \l__penrose_tmpa_fp}
641 }
642 \prop_put:Nnn \g__penrose_pentagon_lms_action_prop {F} {
643   \fp_set:Nn \l__penrose_tmpa_fp { tand(54) * \l__penrose_step_dim }
644   \pgftransformyshift{\fp_to_dim:N \l__penrose_tmpa_fp}
645 }
646 \prop_put:Nnn \g__penrose_pentagon_lms_action_prop {d} {
647   \fp_set:Nn \l__penrose_tmpa_fp
648   {
649     (tand(54)/2 - tand(72)/2 + sind(36) ) * \l__penrose_step_dim
650   }
651   \pgftransformyshift{\fp_to_dim:N \l__penrose_tmpa_fp}
652 }
653 \prop_put:Nnn \g__penrose_pentagon_lms_action_prop {e} {
654   \fp_set:Nn \l__penrose_tmpa_fp
655   {
656     tand(54) * cosd(36) * \l__penrose_step_dim
657   }
658   \pgftransformyshift{\fp_to_dim:N \l__penrose_tmpa_fp}
659 }
660 \prop_put:Nnn \g__penrose_pentagon_lms_action_prop {P} {
661   \group_begin:
662   \int_gincr:N \l__penrose_tile_int
663   \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim/2}
664   \pgftransformxshift{-\fp_to_dim:N \l__penrose_tmpa_fp}
665   \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim*tand(54)/2}
666   \pgftransformyshift{-\fp_to_dim:N \l__penrose_tmpa_fp}
667   \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim/(1cm)}
668   \pgftransformscale{\fp_use:N \l__penrose_tmpa_fp}
669   \tl_set:Nx \l__penrose_tmpe_tl
670   {
671     {\int_use:N \l__penrose_tile_int}
672     {\int_use:N \l__penrose_tiles_int}
673   }
674   \UsePenroseTile[
675     every~ Penrose~ tile/.try,
676     every~ pentagon/.try,
677     every~
678     \seq_item:Nn \l__penrose_pentagon_parity_seq
679     {\l__penrose_pentagon_parity_int}

```

```

680     \space pentagon/.try,
681     every~ pentagon~ 5/.try,
682     Penrose~ tile~ \int_use:N \l__penrose_tile_int/.try,
683     Penrose~ tile/.try/.expand~ once=\l__penrose_tmpe_tl
684 ]{pentagon~5}
685 \group_end:
686 }

687 \prop_put:Nnn \g__penrose_pentagon_lms_action_prop {Q} {
688   \group_begin:
689   \int_gincr:N \l__penrose_tile_int
690   \fp_set:Nn \l__penrose_tmpe_tl {\l__penrose_step_dim/2}
691   \pgftransformxshift{-\fp_to_dim:N \l__penrose_tmpe_tl}
692   \fp_set:Nn \l__penrose_tmpe_tl {\l__penrose_step_dim*tand(54)/2}
693   \pgftransformyshift{-\fp_to_dim:N \l__penrose_tmpe_tl}
694   \fp_set:Nn \l__penrose_tmpe_tl {\l__penrose_step_dim/(1cm)}
695   \pgftransformscale{\fp_use:N \l__penrose_tmpe_tl}
696   \tl_set:Nx \l__penrose_tmpe_tl
697   {
698     {\int_use:N \l__penrose_tile_int}
699     {\int_use:N \l__penrose_tiles_int}
700   }
701   \UsePenroseTile[
702     every~ Penrose~ tile/.try,
703     every~ pentagon/.try,
704     every~
705     \seq_item:Nn \l__penrose_pentagon_parity_seq
706     {\l__penrose_pentagon_parity_int}
707     \space pentagon/.try,
708     every~ pentagon~ 3/.try,
709     Penrose~ tile~ \int_use:N \l__penrose_tile_int/.try,
710     Penrose~ tile/.try/.expand~ once=\l__penrose_tmpe_tl
711 ]{pentagon~3}
712 \group_end:
713 }

714 \prop_put:Nnn \g__penrose_pentagon_lms_action_prop {R} {
715   \group_begin:
716   \int_gincr:N \l__penrose_tile_int
717   \fp_set:Nn \l__penrose_tmpe_tl {\l__penrose_step_dim/2}
718   \pgftransformxshift{-\fp_to_dim:N \l__penrose_tmpe_tl}
719   \fp_set:Nn \l__penrose_tmpe_tl {\l__penrose_step_dim*tand(54)/2}
720   \pgftransformyshift{-\fp_to_dim:N \l__penrose_tmpe_tl}
721   \fp_set:Nn \l__penrose_tmpe_tl {\l__penrose_step_dim/(1cm)}
722   \pgftransformscale{\fp_use:N \l__penrose_tmpe_tl}
723   \tl_set:Nx \l__penrose_tmpe_tl
724   {
725     {\int_use:N \l__penrose_tile_int}
726     {\int_use:N \l__penrose_tiles_int}
727   }
728   \UsePenroseTile[
729     every~ Penrose~ tile/.try,
730     every~ pentagon/.try,
731     every~
732     \seq_item:Nn \l__penrose_pentagon_parity_seq

```

```

733     {\l__penrose_pentagon_parity_int}
734     \space pentagon/.try,
735     every~ pentagon~ 2/.try,
736     Penrose~ tile~ \int_use:N \l__penrose_tile_int/.try,
737     Penrose~ tile/.try/.expand~ once=\l__penrose_tmpc_tl
738 ]{pentagon~2}
739 \group_end:
740 }

741 \prop_put:Nnn \g__penrose_pentagon_lms_action_prop {G} {
742   \group_begin:
743   \int_gincr:N \l__penrose_tile_int
744   % \pgftransformrotate{198}
745   \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim*cosd(72)}
746   \pgftransformxshift{\fp_to_dim:N \l__penrose_tmpa_fp}
747   \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim*tand(54)*cosd(72)}
748   \pgftransformyshift{\fp_to_dim:N \l__penrose_tmpa_fp}
749   \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim/(1cm)}
750   \pgftransformscale{\fp_use:N \l__penrose_tmpa_fp}
751   \tl_set:Nx \l__penrose_tmpc_tl
752   {
753     {\int_use:N \l__penrose_tile_int}
754     {\int_use:N \l__penrose_tiles_int}
755   }
756   \UsePenroseTile[
757     every~ Penrose~ tile/.try,
758     every~ pentagram/.try,
759     Penrose~ tile~ \int_use:N \l__penrose_tile_int/.try,
760     Penrose~ tile/.try/.expand~ once=\l__penrose_tmpc_tl
761 ]{pentagram}
762 \group_end:
763 }

764 \prop_put:Nnn \g__penrose_pentagon_lms_action_prop {B} {
765   \group_begin:
766   \int_gincr:N \l__penrose_tile_int
767   % \pgftransformrotate{198}
768   \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim*cosd(72)}
769   \pgftransformxshift{\fp_to_dim:N \l__penrose_tmpa_fp}
770   \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim*tand(54)*cosd(72)}
771   \pgftransformyshift{\fp_to_dim:N \l__penrose_tmpa_fp}
772   \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim/(1cm)}
773   \pgftransformscale{\fp_use:N \l__penrose_tmpa_fp}
774   \tl_set:Nx \l__penrose_tmpc_tl
775   {
776     {\int_use:N \l__penrose_tile_int}
777     {\int_use:N \l__penrose_tiles_int}
778   }
779   \UsePenroseTile[
780     every~ Penrose~ tile/.try,
781     every~ boat/.try,
782     Penrose~ tile~ \int_use:N \l__penrose_tile_int/.try,
783     Penrose~ tile/.try/.expand~ once=\l__penrose_tmpc_tl
784 ]{boat}
785 \group_end:

```

```

786 }
787 \prop_put:Nnn \g__penrose_pentagon_lms_action_prop {D} {
788   \group_begin:
789   \int_gincr:N \l__penrose_tile_int
790   \pgftransformrotate{90}
791   \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim*cosd(18)}
792   \pgftransformxshift{-\fp_to_dim:N \l__penrose_tmpa_fp}
793   \fp_set:Nn \l__penrose_tmpa_fp {\l__penrose_step_dim/(1cm)}
794   \pgftransformscale{\fp_use:N \l__penrose_tmpa_fp}
795   \tl_set:Nx \l__penrose_tmpc_tl
796   {
797     {\int_use:N \l__penrose_tile_int}
798     {\int_use:N \l__penrose_tiles_int}
799   }
800   \UsePenroseTile[
801     every~ Penrose~ tile/.try,
802     every~ diamond/.try,
803     Penrose~ tile~ \int_use:N \l__penrose_tile_int/.try,
804     Penrose~ tile/.try/.expand~ once=\l__penrose_tmpc_tl
805   ]{diamond}
806   \group_end:
807 }

```

`\penrose_make_lms:Nnnn` This creates the token list of actions, starting with the seed. The arguments are: a token list to store the result in, the name of the system, the number of iterations, and the initial state.

```

808 \cs_new_nopar:Npn \penrose_make_lms:Nnnn #1#2#3#4
809 {
810   \group_begin:

```

On the first time round, we start with the given seed.

```

811   \tl_set:Nn \l__penrose_tmpb_tl {#4}

```

We repeat the specified number of times.

```

812   \prg_replicate:nm {#3} {

```

Duplicate the current state.

```

813     \tl_set_eq:NN \l__penrose_tmpa_tl \l__penrose_tmpb_tl

```

Clear the receiving token list.

```

814     \tl_clear:N \l__penrose_tmpb_tl

```

Walk through the current list, appending to the receiving list according to the rules.

```

815     \tl_map_inline:Nn \l__penrose_tmpa_tl
816     {

```

If a rule exists, copy that.

```

817       \prop_if_in:cnTF {g__penrose_#2_lms_rule_prop} {##1}
818       {
819         \tl_put_right:Nx \l__penrose_tmpb_tl
820         {\prop_item:cn {g__penrose_#2_lms_rule_prop} {##1} }
821       }
822     {

```

Otherwise, just copy the token.

```

823     \tl_put_right:Nn \l__penrose_tmpb_tl {##1}
824   }
825 }
826 }

```

We've done all this inside a group, now pass the result outside.

```

827 \tl_set:Nn \l__penrose_tmpa_tl {
828   \group_end:
829   \tl_set:Nn #1
830 }
831 \tl_put_right:Nx \l__penrose_tmpa_tl {{\tl_use:N \l__penrose_tmpb_tl}}
832 \tl_use:N \l__penrose_tmpa_tl
833 }
834 \cs_generate_variant:Nn \penrose_make_lms:Nnnn {Nnnx}

```

(End definition for \penrose_make_lms:Nnnn.)

`\penrose_invoke_lms:Nn` This carries out the actions specified by the resulting rules.

```

835 \cs_new_nopar:Npn \penrose_invoke_lms:Nn #1#2
836 {
837   \group_begin:

```

Walk through the given list, carrying out the corresponding action if it exists. If not, look at the default. Otherwise, just do nothing.

```

838   \tl_map_inline:Nn #1 {
839     \prop_if_in:cnTF {g__penrose_#2_lms_action_prop} {##1}
840     {
841       \prop_item:cn {g__penrose_#2_lms_action_prop} {##1}
842     }
843     {
844       \prop_if_in:cnT {g__penrose_default_lms_action_prop} {##1}
845       {
846         \prop_item:cn {g__penrose_default_lms_action_prop} {##1}
847       }
848     }
849   }
850   \group_end:
851 }

```

(End definition for \penrose_invoke_lms:Nn.)

We keep track of the number of tiles.

```

852 \int_new:N \l__penrose_tile_int
853 \int_new:N \l__penrose_tiles_int

```

`\PenroseDecomposition` This is the user macro to invoke the decomposition. The arguments are: optional styles, the name, number of iterations, and starting seed.

```

854 \NewDocumentCommand \PenroseDecomposition { O{} m m m }
855 {
856   \group_begin:
857   \tikzset{#1}
858   \penrose_make_lms:Nnnx \l__penrose_tmpa_tl {#2} {#3} {#4}
859   \penrose_count_lms:N \l__penrose_tmpa_tl
860   \int_gzero:N \l__penrose_tile_int

```

```

861 \int_gset:Nn \l__penrose_pentagon_parity_int {2}
862 \penrose_invoke_lms:Nn \l__penrose_tmpa_tl {#2}
863 \group_end:
864 }

```

(End definition for `\PenroseDecomposition`.)

`\penrose_count_lms:N` This counts the number of tiles in the string.

```

865 \cs_new_nopar:Npn \penrose_count_lms:N #1
866 {
867   \int_gzero:N \l__penrose_tiles_int
868   \tl_map_inline:Nn #1
869   {
870     \tl_if_eq:nnT {##1} {T}
871     {
872       \int_incr:N \l__penrose_tiles_int
873     }
874     \tl_if_eq:nnT {##1} {t}
875     {
876       \int_incr:N \l__penrose_tiles_int
877     }
878     \tl_if_eq:nnT {##1} {G}
879     {
880       \int_incr:N \l__penrose_tiles_int
881     }
882     \tl_if_eq:nnT {##1} {g}
883     {
884       \int_incr:N \l__penrose_tiles_int
885     }
886   }
887 }

```

(End definition for `\penrose_count_lms:N`.)

This is a `\tikzset` mechanism for setting the dimensions of the tiling.

```

888 \tikzset{
889   Penrose~ step/.code={
890     \dim_set:Nn \l__penrose_step_dim {#1}
891   }
892 }

```

We're done with \LaTeX 3, so turn off the syntax.

```

893 \ExplSyntaxOff

```

2.5 TikZ Pictures

New in TikZ3.0 is the ability to make pictures that can be reused. Penrose tiles seems an obvious use for this. These pictures can be placed alongside other tiles, matching by edge type.

There are a variety of constants that are frequently used and reused, so we define them all here. These are the PGF versions.

```

894 \pgfmathsetmacro\pr@chphi{cos(18)}
895 \pgfmathsetmacro\pr@shphi{sin(18)}
896 \pgfmathsetmacro\pr@cphi{cos(36)}
897 \pgfmathsetmacro\pr@sphi{sin(36)}

```

```

898 \pgfmathsetmacro\pr@invphi{2/(sqrt(5)+1)}
899 \pgfmathsetmacro\pr@phi{(sqrt(5)+1)/2}
900 \pgfmathsetmacro\pr@invphisq{\pr@invphi*\pr@invphi}
901 \pgfmathsetmacro\pr@ominvphisq{\pr@invphi - \pr@invphisq}
902 \pgfmathsetmacro\pr@ominvphi{1 - \pr@invphi}

903 \newif\if@edgealign
904 \def\pr@getfirst#1#2\pr@stop{#1}%
905 \def\test@edgealign#1{%
906   \pgfkeysgetvalue{/tikz/Penrose/alignment edge}{\@penrose@tmpa}%
907   \edef\@penrose@tmpb{#1}%
908   \edef\@penrose@tmpa{\expandafter\pr@getfirst\@penrose@tmpa.\pr@stop}%
909   \ifx\@penrose@tmpa\@penrose@tmpb
910     \@edgealigntrue
911   \else
912     \@edgealignfalse
913   \fi
914 }%

915 \newif\if@newedge
916 \def\test@newedge#1{%
917   \pgfkeysgetvalue{/tikz/Penrose/alignment new edge}{\@penrose@tmpa}%
918   \edef\@penrose@tmpb{#1}%
919   \edef\@penrose@tmpa{\@penrose@tmpa}%
920   \ifx\@penrose@tmpa\@penrose@tmpb
921     \@newedgetrue
922   \else
923     \@newedgefalse
924   \fi
925 }%

```

The implementation is essentially the same for each, so only the first will be commented.

```

926 \tikzset{

```

The key `align with=<tile> along <edge>` is used to set the parameters for placing a tile next to an existing one. For most tiles, that's enough to specify how the new tile should be placed. Some, though, need more information. For those, use `align with=<tile> along <edge> using <number>`.

```

927   align with/.code args={#1 along #2}{%
928     \pgfutil@in@{ using }{#2}
929     \ifpgfutil@in@%
930       \tikzset{
931         Penrose/alignment location=#1,
932         Penrose/alignment set edges=#2,
933       }%
934     \else
935       \tikzset{
936         Penrose/alignment location=#1,
937         Penrose/alignment edge=#2,
938       }%
939     \fi
940   },
941   Penrose/alignment set edges/.code args={#1 using #2}{%
942     \tikzset{
943       Penrose/alignment edge=#1,

```

```

944     Penrose/alignment new edge=#2
945   },
946 },
947 Penrose/alignment location/.initial={},
948 Penrose/alignment edge/.initial=a,
949 Penrose/alignment new edge/.initial={},

```

Default clipping style.

```

950 every Penrose tile clip/.style={clip},

```

This is the code for setting up a pic.

```

951 thin rhombus/.pic={
952   \begin{scope}

```

Were we given a tile to align ourselves against?

```

953   \pgfkeysgetvalue{/tikz/Penrose/alignment location}{\prloc}
954   \ifx\prloc\pgfutil@empty
955   \else

```

Yes, we were. So we adjust our position accordingly. The first job is to transform so that we're along the edge of the receiving tile.

```

956   \begin{group}

```

We get the locations of the start and end of the receiving tile. As pic sets the node prefix, we have to temporarily suspend that (hence working in a group).

```

957   \tikzset{name prefix ..}%
958   \tikz@scan@one@point\pgfutil@firstofone%
959   (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} start)%
960   \global\pgf@xa=\pgf@x
961   \global\pgf@ya=\pgf@y
962   \tikz@scan@one@point\pgfutil@firstofone%
963   (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} end)%
964   \global\pgf@xb=\pgf@x
965   \global\pgf@yb=\pgf@y
966   \end{group}

```

We store the initial points in `\pgf@xa` and `\pgf@ya` but we want `\pgf@xb` and `\pgf@yb` to be a vector along the edge.

```

967   \advance\pgf@xb by -\pgf@xa
968   \advance\pgf@yb by -\pgf@ya

```

We shift to the start of the edge.

```

969   \pgftransformshift{\pgfpoint{\pgf@xa}{\pgf@ya}}%

```

And normalise the vector along it.

```

970   \pgfpointnormalised{\pgfpoint{\pgf@xb}{\pgf@yb}}
971   \pgf@xb=\pgf@x
972   \pgf@yb=\pgf@y

```

Now rotate so that the x -axis lies along the edge.

```

973   \pgftransformtriangle%
974   {\pgfpoint{0pt}{0pt}}%
975   {\pgfpoint{\pgf@xb}{\pgf@yb}}%
976   {\pgfpoint{-\pgf@yb}{\pgf@xb}}

```


The next job is to shift and rotate the current tile so that the correct edge ends up against the receiving tile.

```

977 \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}b\relax
978 \pgftransformrotate{-18}%
979 \pgftransformshift{\pgfpoint{-\pr@chphi cm}{\pr@shphi cm}}%
980 \else
981 \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}B\relax
982 \pgftransformrotate{18}%
983 \else
984 \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}a\relax
985 \pgftransformrotate{198}%
986 \pgftransformshift{\pgfpoint{-2*\pr@chphi cm}{0 cm}}%
987 \else
988 \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}A\relax
989 \pgftransformrotate{162}%
990 \pgftransformshift{\pgfpoint{-\pr@chphi cm}{-\pr@shphi cm}}%
991 \fi\fi\fi\fi
992 \fi

```

Now that the transformation is finalised, we can render the tile. We clip against the tile path so that the tiles don't "bleed". If we didn't do this, drawing the tile would result in overlaps which can look a bit ugly. On the other hand, tight clipping can lead to "gaps" between the tiles so we make this optional by enclosing it in a style.

```

993 \UsePenroseTile[
994     every Penrose tile clip/.try,
995     every thin rhombus clip/.try
996 ]{thin rhombus}
997 \UsePenroseTile[
998     every Penrose tile/.try,
999     every thin rhombus/.try,
1000     pic actions
1001 ]{thin rhombus}

```

These draw the arcs that designate the joining rules. We draw full circles so that it doesn't matter what shape the tiles are.

```

1002 \UsePenroseTile[
1003     every Penrose arc clip/.try,
1004     every thin rhombus arc clip/.try
1005 ]{thin rhombus}
1006 \path[every circle arc/.try] (18:1) circle[radius=1/4];
1007 \path[every long arc/.try] (-18:1) circle[radius=1/4];

```

Lastly, we put coordinates at each vertex, labelled by which edge they are.

```

1008 \coordinate (-edge a start) at (0,0);
1009 \coordinate (-edge a end) at (18:1);
1010 \coordinate (-edge A start) at (18:1);
1011 \coordinate (-edge A end) at (2*\pr@chphi,0);
1012 \coordinate (-edge B start) at (2*\pr@chphi,0);
1013 \coordinate (-edge B end) at (-18:1);
1014 \coordinate (-edge b start) at (-18:1);
1015 \coordinate (-edge b end) at (0,0);
1016 \end{scope}
1017 },

```

This is a shortcut for installing the pic type.

```

1018 thin rhombus/.style={
1019     every Penrose pic/.try,
1020     pic type=thin rhombus,
1021 },

```

Same again, but for the thick rhombus.

```

1022 thick rhombus/.pic={
1023     \begin{scope}
1024     \pgfkeysgetvalue{/tikz/Penrose/alignment location}{\prloc}
1025     \ifx\prloc\pgfutil@empty
1026     \else
1027     \begin{group}
1028     \tikzset{name prefix ..}%
1029     \tikz@scan@one@point\pgfutil@firstofone%
1030     (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} start)%
1031     \global\pgf@xa=\pgf@x
1032     \global\pgf@ya=\pgf@y
1033     \tikz@scan@one@point\pgfutil@firstofone%
1034     (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} end)%
1035     \global\pgf@xb=\pgf@x
1036     \global\pgf@yb=\pgf@y
1037     \end{group}
1038     \advance\pgf@xb by -\pgf@xa
1039     \advance\pgf@yb by -\pgf@ya
1040     \pgftransformshift{\pgfpoint{\pgf@xa}{\pgf@ya}}%
1041     \pgfpointnormalised{\pgfpoint{\pgf@xb}{\pgf@yb}}
1042     \pgf@xb=\pgf@x
1043     \pgf@yb=\pgf@y
1044     \pgftransformtriangle%
1045     {\pgfpoint{0pt}{0pt}}%
1046     {\pgfpoint{\pgf@xb}{\pgf@yb}}%
1047     {\pgfpoint{-\pgf@yb}{\pgf@xb}}
1048     \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}b\relax
1049     \pgftransformrotate{144}%
1050     \pgftransformshift{\pgfpoint{-\pr@cphi cm}{-\pr@sphi cm}}%
1051     \else
1052     \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}B\relax
1053     \pgftransformrotate{36}%
1054     \else
1055     \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}a\relax
1056     \pgftransformrotate{-36}%
1057     \pgftransformshift{\pgfpoint{-\pr@cphi cm}{\pr@sphi cm}}%
1058     \else
1059     \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}A\relax
1060     \pgftransformrotate{216}%
1061     \pgftransformshift{\pgfpoint{-2*\pr@cphi cm}{0 cm}}%
1062     \fi\fi\fi\fi
1063     \fi
1064     \UsePenroseTile[
1065         every Penrose tile clip/.try,
1066         every thick rhombus clip/.try
1067     ]{thick rhombus}
1068     \UsePenroseTile[

```

```

1069     every Penrose tile/.try,
1070     every thick rhombus/.try,
1071     pic actions
1072   ]{thick rhombus}
1073   \UsePenroseTile[
1074     every Penrose arc clip/.try,
1075     every thick rhombus arc clip/.try
1076   ]{thick rhombus}
1077   \path[every circle arc/.try] (2*\pr@cphi,0) circle[radius=1/4];
1078   \path[every long arc/.try] (0,0) circle[radius=3/4];
1079   \coordinate (-edge B start) at (0,0);
1080   \coordinate (-edge B end) at (36:1);
1081   \coordinate (-edge a start) at (36:1);
1082   \coordinate (-edge a end) at (2*\pr@cphi,0);
1083   \coordinate (-edge A start) at (2*\pr@cphi,0);
1084   \coordinate (-edge A end) at (-36:1);
1085   \coordinate (-edge b start) at (-36:1);
1086   \coordinate (-edge b end) at (0,0);
1087   \end{scope}
1088 },
1089 thick rhombus/.style={
1090   every Penrose pic/.try,
1091   pic type=thick rhombus,
1092 },

```

Now the kite.

```

1093 kite/.pic={
1094   \begin{scope}
1095     \pgfkeysgetvalue{/tikz/Penrose/alignment location}{\prloc}
1096     \ifx\prloc\pgfutil@empty
1097       \else
1098         \begingroup
1099         \tikzset{name prefix ..}%
1100         \tikz@scan@one@point\pgfutil@firstofone%
1101         (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} start)%
1102         \global\pgf@xa=\pgf@x
1103         \global\pgf@ya=\pgf@y
1104         \tikz@scan@one@point\pgfutil@firstofone%
1105         (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} end)%
1106         \global\pgf@xb=\pgf@x
1107         \global\pgf@yb=\pgf@y
1108         \endgroup
1109         \advance\pgf@xb by -\pgf@xa
1110         \advance\pgf@yb by -\pgf@ya
1111         \pgftransformshift{\pgfpoint{\pgf@xa}{\pgf@ya}}%
1112         \pgfpointnormalised{\pgfpoint{\pgf@xb}{\pgf@yb}}
1113         \pgf@xb=\pgf@x
1114         \pgf@yb=\pgf@y
1115         \pgftransformtriangle%
1116         {\pgfpoint{0pt}{0pt}}%
1117         {\pgfpoint{\pgf@xb}{\pgf@yb}}%
1118         {\pgfpoint{-\pgf@yb}{\pgf@xb}}%
1119         \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}c\relax
1120         \pgftransformrotate{-72}%
1121         \pgftransformshift{\pgfpoint{-\pr@cphi cm}{\pr@sphi cm}}%

```

```

1122 \else
1123 \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}C\relax
1124 \pgftransformrotate{-108}%
1125 \pgftransformshift{\pgfpoint{-1 cm}{0 cm}}%
1126 \else
1127 \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}a\relax
1128 \pgftransformrotate{36}%
1129 \else
1130 \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}A\relax
1131 \pgftransformrotate{144}%
1132 \pgftransformshift{\pgfpoint{-\pr@cphi cm}{-\pr@sphi cm}}%
1133 \fi\fi\fi\fi
1134 \fi
1135 \UsePenroseTile[
1136     every Penrose tile clip/.try,
1137     every kite clip/.try
1138 ]{kite}
1139 \UsePenroseTile[
1140     every Penrose tile/.try,
1141     every kite/.try,
1142     pic actions
1143 ]{kite}
1144 \UsePenroseTile[
1145     every Penrose arc clip/.try,
1146     every kite arc clip/.try
1147 ]{kite}
1148 \path[every circle arc/.try] (0,0) circle[radius=\pr@invphi];
1149 \path[every long arc/.try] (1,0) circle[radius=\pr@invphisq];
1150 \coordinate (-edge a start) at (0,0);
1151 \coordinate (-edge a end) at (36:1);
1152 \coordinate (-edge c start) at (36:1);
1153 \coordinate (-edge c end) at (1,0);
1154 \coordinate (-edge C start) at (1,0);
1155 \coordinate (-edge C end) at (-36:1);
1156 \coordinate (-edge A start) at (-36:1);
1157 \coordinate (-edge A end) at (0,0);
1158 \end{scope}
1159 },

```

The dart is next.

```

1160 dart/.pic={
1161     \begin{scope}
1162         \pgfkeysgetvalue{/tikz/Penrose/alignment location}{\prloc}
1163         \ifx\prloc\pgfutil@empty
1164             \else
1165             \begin{group}
1166                 \tikzset{name prefix ..}%
1167                 \tikz@scan@one@point\pgfutil@firstofone%
1168                 (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} start)%
1169                 \global\pgf@xa=\pgf@x
1170                 \global\pgf@ya=\pgf@y
1171                 \tikz@scan@one@point\pgfutil@firstofone%
1172                 (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} end)%
1173                 \global\pgf@xb=\pgf@x
1174                 \global\pgf@yb=\pgf@y

```

```

1175 \endgroup
1176 \advance\pgf@xb by -\pgf@xa
1177 \advance\pgf@yb by -\pgf@ya
1178 \pgftransformshift{\pgfpoint{\pgf@xa}{\pgf@ya}}%
1179 \pgfpointnormalised{\pgfpoint{\pgf@xb}{\pgf@yb}}
1180 \pgf@xb=\pgf@x
1181 \pgf@yb=\pgf@y
1182 \pgftransformtriangle%
1183 {\pgfpoint{0pt}{0pt}}%
1184 {\pgfpoint{\pgf@xb}{\pgf@yb}}%
1185 {\pgfpoint{-\pgf@yb}{\pgf@xb}}%
1186 \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}c\relax
1187 \pgftransformrotate{108}%
1188 \else
1189 \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}C\relax
1190 \pgftransformrotate{72}%
1191 \pgftransformshift%
1192 {\pgfpoint{\pr@invphi*\pr@shphi cm}{-\pr@invphi*\pr@chphi cm}}%
1193 \else
1194 \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}a\relax
1195 \pgftransformrotate{-36}%
1196 \pgftransformshift%
1197 {\pgfpoint{\pr@invphi*\pr@shphi cm}{\pr@invphi*\pr@chphi cm}}%
1198 \else
1199 \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}A\relax
1200 \pgftransformrotate{216}%
1201 \pgftransformshift{\pgfpoint{-\pr@invphi cm}{0 cm}}%
1202 \fi\fi\fi\fi
1203 \fi
1204 \UsePenroseTile[
1205   every Penrose tile clip/.try,
1206   every dart clip/.try
1207 ]{dart}
1208 \UsePenroseTile[
1209   every Penrose tile/.try,
1210   every dart/.try,
1211   pic actions
1212 ]{dart}
1213 \UsePenroseTile[
1214   every Penrose arc clip/.try,
1215   every dart arc clip/.try
1216 ]{dart}
1217 \path[every circle arc/.try] (\pr@invphi,0) circle[radius=\pr@ominvphi];
1218 \path[every long arc/.try] (0,0) circle[radius=\pr@ominvphisq];
1219 \coordinate (-edge c start) at (0,0);
1220 \coordinate (-edge c end) at (108:\pr@invphi);
1221 \coordinate (-edge a start) at (108:\pr@invphi);
1222 \coordinate (-edge a end) at (\pr@invphi,0);
1223 \coordinate (-edge A start) at (\pr@invphi,0);
1224 \coordinate (-edge A end) at (-108:\pr@invphi);
1225 \coordinate (-edge C start) at (-108:\pr@invphi);
1226 \coordinate (-edge C end) at (0,0);
1227 \end{scope}
1228 },

```

```

1229 kite/.style={
1230   every Penrose pic/.try,
1231   pic type=kite,
1232 },
1233 dart/.style={
1234   every Penrose pic/.try,
1235   pic type=dart,
1236 },

```

The golden triangle.

```

1237 golden triangle/.pic={
1238   \begin{scope}
1239     \pgfkeysgetvalue{/tikz/Penrose/alignment location}{\prloc}
1240     \ifx\prloc\pgfutil@empty
1241     \else
1242     \begingroup
1243     \tikzset{name prefix ..}%
1244     \tikz@scan@one@point\pgfutil@firstofone%
1245     (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} start)%
1246     \global\pgf@xa=\pgf@x
1247     \global\pgf@ya=\pgf@y
1248     \tikz@scan@one@point\pgfutil@firstofone%
1249     (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} end)%
1250     \global\pgf@xb=\pgf@x
1251     \global\pgf@yb=\pgf@y
1252     \endgroup
1253     \advance\pgf@xb by -\pgf@xa
1254     \advance\pgf@yb by -\pgf@ya
1255     \pgftransformshift{\pgfpoint{\pgf@xa}{\pgf@ya}}%
1256     \pgfpointnormalised{\pgfpoint{\pgf@xb}{\pgf@yb}}
1257     \pgf@xb=\pgf@x
1258     \pgf@yb=\pgf@y
1259     \pgftransformtriangle%
1260     {\pgfpoint{0pt}{0pt}}%
1261     {\pgfpoint{\pgf@xb}{\pgf@yb}}%
1262     {\pgfpoint{-\pgf@yb}{\pgf@xb}}%
1263     \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}B\relax
1264     \pgftransformrotate{18}%
1265     \else
1266     \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}C\relax
1267     \pgftransformrotate{-90}%
1268     \pgftransformshift{\pgfpoint{-\pr@chphi cm}{\pr@shphi cm}}%
1269     \else
1270     \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}A\relax
1271     \pgftransformrotate{162}%
1272     \pgftransformshift{\pgfpoint{-\pr@chphi cm}{-\pr@shphi cm}}%
1273     \fi\fi\fi
1274     \fi
1275     \UsePenroseTile[
1276       every Penrose tile clip/.try,
1277       every golden triangle clip/.try
1278     ]{golden triangle}
1279     \UsePenroseTile[
1280       every Penrose tile/.try,
1281       every golden triangle/.try,

```

```

1282     pic actions
1283     ]{golden triangle}
1284 \coordinate (-edge a start) at (0,0);
1285 \coordinate (-edge a end) at (18:1);
1286 \coordinate (-edge c start) at (18:1);
1287 \coordinate (-edge c end) at (-18:1);
1288 \coordinate (-edge b start) at (-18:1);
1289 \coordinate (-edge b end) at (0,0);
1290 \end{scope}
1291 },
1292 golden triangle/.style={
1293     every Penrose pic/.try,
1294     pic type=golden triangle,
1295 },

```

The reverse golden triangle (is there a better name?).

```

1296 reverse golden triangle/.pic={
1297     \begin{scope}
1298     \pgfkeysgetvalue{/tikz/Penrose/alignment location}{\prloc}
1299     \ifx\prloc\pgfutil@empty
1300     \else
1301     \begin{group}
1302     \tikzset{name prefix ..}%
1303     \tikz@scan@one@point\pgfutil@firstofone%
1304     (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} start)%
1305     \global\pgf@xa=\pgf@x
1306     \global\pgf@ya=\pgf@y
1307     \tikz@scan@one@point\pgfutil@firstofone%
1308     (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} end)%
1309     \global\pgf@xb=\pgf@x
1310     \global\pgf@yb=\pgf@y
1311     \end{group}
1312     \advance\pgf@xb by -\pgf@xa
1313     \advance\pgf@yb by -\pgf@ya
1314     \pgftransformshift{\pgfpoint{\pgf@xa}{\pgf@ya}}%
1315     \pgfpointnormalised{\pgfpoint{\pgf@xb}{\pgf@yb}}
1316     \pgf@xb=\pgf@x
1317     \pgf@yb=\pgf@y
1318     \pgftransformtriangle%
1319     {\pgfpoint{0pt}{0pt}}%
1320     {\pgfpoint{\pgf@xb}{\pgf@yb}}%
1321     {\pgfpoint{-\pgf@yb}{\pgf@xb}}%
1322     \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}b\relax
1323     \pgftransformrotate{162}%
1324     \pgftransformshift{\pgfpoint{-\pr@chphi cm}{-\pr@shphi cm}}%
1325     \else
1326     \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}c\relax
1327     \pgftransformrotate{-90}%
1328     \pgftransformshift{\pgfpoint{-\pr@chphi cm}{\pr@shphi cm}}%
1329     \else
1330     \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}a\relax
1331     \pgftransformrotate{18}%
1332     \fi\fi\fi
1333     \fi
1334     \UsePenroseTile[

```

```

1335     every Penrose tile clip/.try,
1336     every reverse golden triangle clip/.try
1337 ]{reverse golden triangle}
1338 \UsePenroseTile[
1339     every Penrose tile/.try,
1340     every reverse golden triangle/.try,
1341     pic actions
1342 ]{reverse golden triangle}
1343 \coordinate (-edge B start) at (0,0);
1344 \coordinate (-edge B end) at (18:1);
1345 \coordinate (-edge C start) at (18:1);
1346 \coordinate (-edge C end) at (-18:1);
1347 \coordinate (-edge A start) at (-18:1);
1348 \coordinate (-edge A end) at (0,0);
1349 \end{scope}
1350 },
1351 reverse golden triangle/.style={
1352     every Penrose pic/.try,
1353     pic type=reverse golden triangle,
1354 },

```

The golden gnomon.

```

1355 golden gnomon/.pic={
1356     \begin{scope}
1357     \pgfkeysgetvalue{/tikz/Penrose/alignment location}{\prloc}
1358     \ifx\prloc\pgfutil@empty
1359     \else
1360     \begin{group}
1361     \tikzset{name prefix ..}%
1362     \tikz@scan@one@point\pgfutil@firstofone%
1363     (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} start)%
1364     \global\pgf@xa=\pgf@x
1365     \global\pgf@ya=\pgf@y
1366     \tikz@scan@one@point\pgfutil@firstofone%
1367     (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} end)%
1368     \global\pgf@xb=\pgf@x
1369     \global\pgf@yb=\pgf@y
1370     \end{group}
1371     \advance\pgf@xb by -\pgf@xa
1372     \advance\pgf@yb by -\pgf@ya
1373     \pgftransformshift{\pgfpoint{\pgf@xa}{\pgf@ya}}%
1374     \pgfpointnormalised{\pgfpoint{\pgf@xb}{\pgf@yb}}
1375     \pgf@xb=\pgf@x
1376     \pgf@yb=\pgf@y
1377     \pgftransformtriangle%
1378     {\pgfpoint{0pt}{0pt}}%
1379     {\pgfpoint{\pgf@xb}{\pgf@yb}}%
1380     {\pgfpoint{-\pgf@yb}{\pgf@xb}}%
1381     \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}c\relax
1382     \pgftransformrotate{144}%
1383     \pgftransformshift{\pgfpoint{-\pr@cphi cm}{-\pr@sphi cm}}%
1384     \else
1385     \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}B\relax
1386     \pgftransformrotate{-144}%
1387     \pgftransformshift{\pgfpoint{-2*\pr@cphi cm}{0 cm}}%

```



```

1388 \else
1389 \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}a\relax
1390 \fi\fi\fi
1391 \fi
1392 \UsePenroseTile[
1393   every Penrose tile clip/.try,
1394   every golden gnomon clip/.try
1395 ]{golden gnomon}
1396 \UsePenroseTile[
1397   every Penrose tile/.try,
1398   every golden gnomon/.try,
1399   pic actions
1400 ]{golden gnomon}
1401 \coordinate (-edge C start) at (0,0);
1402 \coordinate (-edge C end) at (36:1);
1403 \coordinate (-edge b start) at (36:1);
1404 \coordinate (-edge b end) at (2*\pr@cphi,0);
1405 \coordinate (-edge A start) at (2*\pr@cphi,0);
1406 \coordinate (-edge A end) at (0,0);
1407 \end{scope}
1408 },
1409 golden gnomon/.style={
1410   every Penrose pic/.try,
1411   pic type=golden gnomon,
1412 },

```

The reverse golden gnomon.

```

1413 reverse golden gnomon/.pic={
1414   \begin{scope}
1415   \pgfkeysgetvalue{/tikz/Penrose/alignment location}{\prloc}
1416   \ifx\prloc\pgfutil@empty
1417   \else
1418   \begingroup
1419   \tikzset{name prefix ..}%
1420   \tikz@scan@one@point\pgfutil@firstofone%
1421   (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} start)%
1422   \global\pgf@xa=\pgf@x
1423   \global\pgf@ya=\pgf@y
1424   \tikz@scan@one@point\pgfutil@firstofone%
1425   (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} end)%
1426   \global\pgf@xb=\pgf@x
1427   \global\pgf@yb=\pgf@y
1428   \endgroup
1429   \advance\pgf@xb by -\pgf@xa
1430   \advance\pgf@yb by -\pgf@ya
1431   \pgftransformshift{\pgfpoint{\pgf@xa}{\pgf@ya}}%
1432   \pgfpointnormalised{\pgfpoint{\pgf@xb}{\pgf@yb}}
1433   \pgf@xb=\pgf@x
1434   \pgf@yb=\pgf@y
1435   \pgftransformtriangle%
1436   {\pgfpoint{0pt}{0pt}}%
1437   {\pgfpoint{\pgf@xb}{\pgf@yb}}%
1438   {\pgfpoint{-\pgf@yb}{\pgf@xb}}%
1439   \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}C\relax
1440   \pgftransformrotate{36}%

```

```

1441 \else
1442 \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}b\relax
1443 \pgftransformrotate{-36}%
1444 \pgftransformshift{\pgfpoint{-\pr@cphi cm}{\pr@sphi cm}}%
1445 \else
1446 \if\pgfkeysvalueof{/tikz/Penrose/alignment edge}A\relax
1447 \pgftransformrotate{180}%
1448 \pgftransformshift{\pgfpoint{-2*\pr@cphi cm}{0 cm}}%
1449 \fi\fi\fi
1450 \fi
1451 \UsePenroseTile[
1452     every Penrose tile clip/.try,
1453     every reverse golden gnomon clip/.try
1454 ]{reverse golden gnomon}
1455 \UsePenroseTile[
1456     every Penrose tile/.try,
1457     every reverse golden gnomon/.try,
1458     pic actions
1459 ]{reverse golden gnomon}
1460 \coordinate (-edge a start) at (0,0);
1461 \coordinate (-edge a end) at (2*\pr@cphi,0);
1462 \coordinate (-edge B start) at (2*\pr@cphi,0);
1463 \coordinate (-edge B end) at (-36:1);
1464 \coordinate (-edge c start) at (-36:1);
1465 \coordinate (-edge c end) at (0,0);
1466 \end{scope}
1467 },
1468 reverse golden gnomon/.style={
1469     every Penrose pic/.try,
1470     pic type=reverse golden gnomon,
1471 },

```

The primary pentagon.

```

1472 pentagon 5/.pic={
1473     \begin{scope}
1474         \pgfkeysgetvalue{/tikz/Penrose/alignment location}{\prloc}
1475         \ifx\prloc\pgfutil@empty
1476         \else
1477         \begin{group}
1478             \tikzset{name prefix ..}%
1479             \tikz@scan@one@point\pgfutil@firstofone%
1480             (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} start)%
1481             \global\pgf@xa=\pgf@x
1482             \global\pgf@ya=\pgf@y
1483             \tikz@scan@one@point\pgfutil@firstofone%
1484             (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} end)%
1485             \global\pgf@xb=\pgf@x
1486             \global\pgf@yb=\pgf@y
1487         \end{group}
1488         \advance\pgf@xb by -\pgf@xa
1489         \advance\pgf@yb by -\pgf@ya
1490         \pgftransformshift{\pgfpoint{\pgf@xa}{\pgf@ya}}%
1491         \pgfpointnormalised{\pgfpoint{\pgf@xb}{\pgf@yb}}
1492         \pgf@xb=\pgf@x
1493         \pgf@yb=\pgf@y

```

```

1494 \pgftransformtriangle%
1495 {\pgfpoint{0pt}{0pt}}%
1496 {\pgfpoint{\pgf@xb}{\pgf@yb}}%
1497 {\pgfpoint{-\pgf@yb}{\pgf@xb}}%
1498 \test@newedge{1}%
1499 \if@newedge
1500 \pgftransformrotate{180}%
1501 \pgftransformshift{\pgfpoint{-1 cm}{0 cm}}%
1502 \else
1503 \test@newedge{2}%
1504 \if@newedge
1505 \pgftransformshift{\pgfpoint{1 cm}{0 cm}}%
1506 \pgftransformrotate{108}%
1507 \pgftransformshift{\pgfpoint{-1 cm}{0 cm}}%
1508 \else
1509 \test@newedge{3}%
1510 \if@newedge
1511 \pgftransformshift%
1512 {\pgfpoint{\pr@phi * \pr@shphi cm}{-\pr@phi * \pr@chphi cm}}%
1513 \pgftransformrotate{36}%
1514 \else
1515 \test@newedge{4}%
1516 \if@newedge
1517 \pgftransformshift{\pgfpoint{-\pr@shphi cm}{-\pr@chphi cm}}%
1518 \pgftransformrotate{-36}%
1519 \else
1520 \test@newedge{5}%
1521 \if@newedge
1522 \pgftransformrotate{-108}%
1523 \fi\fi\fi\fi\fi
1524 \fi
1525 \UsePenroseTile[
1526   every Penrose tile clip/.try,
1527   every pentagon clip/.try,
1528   every pentagon 5 clip/.try
1529 ]{pentagon 5}
1530 \UsePenroseTile[
1531   every Penrose tile/.try,
1532   every pentagon/.try,
1533   every pentagon 5/.try,
1534   pic actions
1535 ]{pentagon 5}
1536 \coordinate (-edge a1 start) at (0,0);
1537 \coordinate (-edge a1 end) at (1,0);
1538 \coordinate (-edge a2 start) at (1,0);
1539 \coordinate (-edge a2 end) at (1+\pr@shphi,\pr@chphi);
1540 \coordinate (-edge a3 start) at (1+\pr@shphi,\pr@chphi);
1541 \coordinate (-edge a3 end) at (\pr@cphi-\pr@shphi,\pr@sphi+\pr@chphi);
1542 \coordinate (-edge a4 start) at (\pr@cphi-\pr@shphi,\pr@sphi+\pr@chphi);
1543 \coordinate (-edge a4 end) at (-\pr@shphi,\pr@chphi);
1544 \coordinate (-edge a5 start) at (-\pr@shphi,\pr@chphi);
1545 \coordinate (-edge a5 end) at (0,0);
1546 \end{scope}
1547 },

```

```

1548 pentagon 5/.style={
1549     every Penrose pic/.try,
1550     pic type=pentagon 5,
1551 },

```

The secondary pentagon.

```

1552 pentagon 3/.pic={
1553     \begin{scope}
1554     \pgfkeysgetvalue{/tikz/Penrose/alignment location}{\prloc}
1555     \ifx\prloc\pgfutil@empty
1556     \else
1557     \begin{group}
1558     \tikzset{name prefix ..}%
1559     \tikz@scan@one@point\pgfutil@firstofone%
1560     (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} start)%
1561     \global\pgf@xa=\pgf@x
1562     \global\pgf@ya=\pgf@y
1563     \tikz@scan@one@point\pgfutil@firstofone%
1564     (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} end)%
1565     \global\pgf@xb=\pgf@x
1566     \global\pgf@yb=\pgf@y
1567     \end{group}
1568     \advance\pgf@xb by -\pgf@xa
1569     \advance\pgf@yb by -\pgf@ya
1570     \pgftransformshift{\pgfpoint{\pgf@xa}{\pgf@ya}}%
1571     \pgfpointnormalised{\pgfpoint{\pgf@xb}{\pgf@yb}}
1572     \pgf@xb=\pgf@x
1573     \pgf@yb=\pgf@y
1574     \pgftransformtriangle%
1575     {\pgfpoint{0pt}{0pt}}%
1576     {\pgfpoint{\pgf@xb}{\pgf@yb}}%
1577     {\pgfpoint{-\pgf@yb}{\pgf@xb}}%
1578     \test@edgealign{a}%
1579     \if@edgealign
1580     \pgftransformrotate{180}%
1581     \pgftransformshift{\pgfpoint{-1 cm}{0 cm}}%
1582     \else
1583     \test@edgealign{B}%
1584     \if@edgealign%
1585     \test@newedge{1}%
1586     \if@newedge
1587     \pgftransformshift{\pgfpoint{1 cm}{0 cm}}%
1588     \pgftransformrotate{108}%
1589     \pgftransformshift{\pgfpoint{-1 cm}{0 cm}}%
1590     \else
1591     \pgftransformrotate{-108}%
1592     \fi
1593     \else
1594     \test@newedge{1}%
1595     \if@newedge
1596     \pgftransformshift%
1597     {\pgfpoint{\pr@phi * \pr@shphi cm}{-\pr@phi * \pr@chphi cm}}%
1598     \pgftransformrotate{36}%
1599     \else
1600     \pgftransformshift{\pgfpoint{-\pr@shphi cm}{-\pr@chphi cm}}%

```

```

1601 \pgftransformrotate{-36}%
1602 \fi\fi\fi
1603 \fi
1604 \UsePenroseTile[
1605     every Penrose tile clip/.try,
1606     every pentagon clip/.try,
1607     every pentagon 3 clip/.try
1608 ]{pentagon 3}
1609 \UsePenroseTile[
1610     every Penrose tile/.try,
1611     every pentagon/.try,
1612     every pentagon 3/.try,
1613     pic actions
1614 ]{pentagon 3}
1615 \coordinate (-edge A start) at (0,0);
1616 \coordinate (-edge A end) at (1,0);
1617 \coordinate (-edge b1 start) at (1,0);
1618 \coordinate (-edge b1 end) at (1+\pr@shphi,\pr@chphi);
1619 \coordinate (-edge a1 start) at (1+\pr@shphi,\pr@chphi);
1620 \coordinate (-edge a1 end) at (\pr@cphi-\pr@shphi,\pr@sphi+\pr@chphi);
1621 \coordinate (-edge a2 start) at (\pr@cphi-\pr@shphi,\pr@sphi+\pr@chphi);
1622 \coordinate (-edge a2 end) at (-\pr@shphi,\pr@chphi);
1623 \coordinate (-edge b2 start) at (-\pr@shphi,\pr@chphi);
1624 \coordinate (-edge b2 end) at (0,0);
1625 \end{scope}
1626 },
1627 pentagon 3/.style={
1628     every Penrose pic/.try,
1629     pic type=pentagon 3,
1630 },

```

The tertiary pentagon.

```

1631 pentagon 2/.pic={
1632     \begin{scope}
1633         \pgfkeysgetvalue{/tikz/Penrose/alignment location}{\prloc}
1634         \ifx\prloc\pgfutil@empty
1635             \else
1636             \begingroup
1637                 \tikzset{name prefix ..}%
1638                 \tikz@scan@one@point\pgfutil@firstofone%
1639                 (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} start)%
1640                 \global\pgf@xa=\pgf@x
1641                 \global\pgf@ya=\pgf@y
1642                 \tikz@scan@one@point\pgfutil@firstofone%
1643                 (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} end)%
1644                 \global\pgf@xb=\pgf@x
1645                 \global\pgf@yb=\pgf@y
1646             \endgroup
1647             \advance\pgf@xb by -\pgf@xa
1648             \advance\pgf@yb by -\pgf@ya
1649             \pgftransformshift{\pgfpoint{\pgf@xa}{\pgf@ya}}%
1650             \pgfpointnormalised{\pgfpoint{\pgf@xb}{\pgf@yb}}
1651             \pgf@xb=\pgf@x
1652             \pgf@yb=\pgf@y
1653             \pgftransformtriangle%

```

```

1654     {\pgfpoint{0pt}{0pt}}%
1655     {\pgfpoint{\pgf@xb}{\pgf@yb}}%
1656     {\pgfpoint{-\pgf@yb}{\pgf@xb}}%
1657     \test@edgealign{D}%
1658     \if@edgealign
1659     \pgftransformrotate{180}%
1660     \pgftransformshift{\pgfpoint{-1 cm}{0 cm}}%
1661     \else
1662     \test@edgealign{a}%
1663     \if@edgealign%
1664     \test@newedge{1}%
1665     \if@newedge
1666     \pgftransformshift{\pgfpoint{1 cm}{0 cm}}%
1667     \pgftransformrotate{108}%
1668     \pgftransformshift{\pgfpoint{-1 cm}{0 cm}}%
1669     \else
1670     \pgftransformrotate{-108}%
1671     \fi
1672     \else
1673     \test@newedge{1}%
1674     \if@newedge
1675     \pgftransformshift%
1676     {\pgfpoint{\pr@phi * \pr@shphi cm}{-\pr@phi * \pr@chphi cm}}%
1677     \pgftransformrotate{36}%
1678     \else
1679     \pgftransformshift{\pgfpoint{-\pr@shphi cm}{-\pr@chphi cm}}%
1680     \pgftransformrotate{-36}%
1681     \fi\fi\fi
1682     \fi
1683     \UsePenroseTile[
1684         every Penrose tile clip/.try,
1685         every pentagon clip/.try,
1686         every pentagon 2 clip/.try
1687     ]{pentagon 2}
1688     \UsePenroseTile[
1689         every Penrose tile/.try,
1690         every pentagon/.try,
1691         every pentagon 2/.try,
1692         pic actions
1693     ]{pentagon 2}
1694     \coordinate (-edge d start) at (0,0);
1695     \coordinate (-edge d end) at (1,0);
1696     \coordinate (-edge A1 start) at (1,0);
1697     \coordinate (-edge A1 end) at (1+\pr@shphi,\pr@chphi);
1698     \coordinate (-edge c1 start) at (1+\pr@shphi,\pr@chphi);
1699     \coordinate (-edge c1 end) at (\pr@cphi-\pr@shphi,\pr@sphi+\pr@chphi);
1700     \coordinate (-edge c2 start) at (\pr@cphi-\pr@shphi,\pr@sphi+\pr@chphi);
1701     \coordinate (-edge c2 end) at (-\pr@shphi,\pr@chphi);
1702     \coordinate (-edge A2 start) at (-\pr@shphi,\pr@chphi);
1703     \coordinate (-edge A2 end) at (0,0);
1704     \end{scope}
1705     },
1706     pentagon 2/.style={
1707         every Penrose pic/.try,

```

```

1708     pic type=pentagon 2,
1709 },

```

The pentagram.

```

1710 pentagram/.pic={
1711   \begin{scope}
1712   \pgfkeysgetvalue{/tikz/Penrose/alignment location}{\prloc}
1713   \ifx\prloc\pgfutil@empty
1714   \else
1715   \begingroup
1716   \tikzset{name prefix ..}%
1717   \tikz@scan@one@point\pgfutil@firstofone%
1718   (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} start)%
1719   \global\pgf@xa=\pgf@x
1720   \global\pgf@ya=\pgf@y
1721   \tikz@scan@one@point\pgfutil@firstofone%
1722   (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} end)%
1723   \global\pgf@xb=\pgf@x
1724   \global\pgf@yb=\pgf@y
1725   \endgroup
1726   \advance\pgf@xb by -\pgf@xa
1727   \advance\pgf@yb by -\pgf@ya
1728   \pgftransformshift{\pgfpoint{\pgf@xa}{\pgf@ya}}%
1729   \pgfpointnormalised{\pgfpoint{\pgf@xb}{\pgf@yb}}
1730   \pgf@xb=\pgf@x
1731   \pgf@yb=\pgf@y
1732   \pgftransformtriangle%
1733   {\pgfpoint{0pt}{0pt}}%
1734   {\pgfpoint{\pgf@xb}{\pgf@yb}}%
1735   {\pgfpoint{-\pgf@yb}{\pgf@xb}}%
1736   \test@newedge{2}%
1737   \if@newedge
1738   \pgftransformshift{\pgfpoint{1 cm}{0 cm}}%
1739   \pgftransformrotate{72}%
1740   \else
1741   \test@newedge{3}%
1742   \if@newedge
1743   \pgftransformrotate{-72}%
1744   \pgftransformshift{\pgfpoint{2 * \pr@shphi cm}{0 cm}}%
1745   \else
1746   \test@newedge{4}%
1747   \if@newedge
1748   \pgftransformshift{\pgfpoint{1 cm + 2 * \pr@shphi cm}{0 cm}}%
1749   \else
1750   \test@newedge{5}%
1751   \if@newedge
1752   \pgftransformrotate{216}%
1753   \pgftransformshift{\pgfpoint{\pr@cphi cm}{\pr@sphi cm}}%
1754   \else
1755   \test@newedge{6}%
1756   \if@newedge
1757   \pgftransformshift{\pgfpoint{1cm}{0cm}}%
1758   \pgftransformrotate{-72}%
1759   \pgftransformshift{\pgfpoint{\pr@cphi cm}{\pr@sphi cm}}%
1760   \else

```

```

1761 \test@newedge{7}%
1762 \if@newedge
1763 \pgftransformrotate{144}%
1764 \pgftransformshift{\pgfpoint{\pr@shphi cm}{\pr@chphi cm}}%
1765 \else
1766 \test@newedge{8}%
1767 \if@newedge
1768 \pgftransformshift{\pgfpoint{1 cm + 2*\pr@shphi cm}{0cm}}%
1769 \pgftransformrotate{216}%
1770 \pgftransformshift{\pgfpoint{\pr@cphi cm}{\pr@sphi cm}}%
1771 \else
1772 \test@newedge{9}%
1773 \if@newedge
1774 \pgftransformshift{\pgfpoint{-2*\pr@shphi cm}{0cm}}%
1775 \pgftransformrotate{72}%
1776 \else
1777 \test@newedge{10}%
1778 \if@newedge
1779 \pgftransformrotate{144}%
1780 \pgftransformshift{\pgfpoint{-1 cm}{0 cm}}%
1781 \else
1782 \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi
1783 \fi
1784 \UsePenroseTile[
1785     every Penrose tile clip/.try,
1786     every pentagram clip/.try
1787 ]{pentagram}
1788 \UsePenroseTile[
1789     every Penrose tile/.try,
1790     every pentagram/.try,
1791     pic actions
1792 ]{pentagram}
1793 \coordinate (-edge C1 start) at (1,0);
1794 \coordinate (-edge C1 end) at (0,0);
1795 \coordinate (-edge C2 start) at (0,0);
1796 \coordinate (-edge C2 end) at (-\pr@shphi,\pr@chphi);
1797 \coordinate (-edge C3 start) at (-\pr@shphi,\pr@chphi);
1798 \coordinate (-edge C3 end) at (-2*\pr@shphi,0);
1799 \coordinate (-edge C4 start) at (-2*\pr@shphi,0);
1800 \coordinate (-edge C4 end) at (-1-2*\pr@shphi,0);
1801 \coordinate (-edge C5 start) at (-1-2*\pr@shphi,0);
1802 \coordinate (-edge C5 end) at (-\pr@cphi,-\pr@sphi);
1803 \coordinate (-edge C6 start) at (-\pr@cphi,-\pr@sphi);
1804 \coordinate (-edge C6 end) at (-\pr@cphi-\pr@shphi,-\pr@sphi-\pr@chphi);
1805 \coordinate (-edge C7 start) at (-\pr@cphi-\pr@shphi,-\pr@sphi-\pr@chphi);
1806 \coordinate (-edge C7 end) at (-\pr@shphi,-\pr@chphi);
1807 \coordinate (-edge C8 start) at (-\pr@shphi,-\pr@chphi);
1808 \coordinate (-edge C8 end) at (1-\pr@cphi+\pr@shphi,-\pr@sphi-\pr@chphi);
1809 \coordinate (-edge C9 start) at (1-\pr@cphi+\pr@shphi,-\pr@sphi-\pr@chphi);
1810 \coordinate (-edge C9 end) at (1-\pr@cphi,-\pr@sphi);
1811 \coordinate (-edge C10 start) at (1-\pr@cphi,-\pr@sphi);
1812 \coordinate (-edge C10 end) at (1,0);
1813 \end{scope}
1814 },

```



```

1815 pentagram/.style={
1816   every Penrose pic/.try,
1817   pic type=pentagram,
1818 },

```

The boat.

```

1819 boat/.pic={
1820   \begin{scope}
1821     \pgfkeysgetvalue{/tikz/Penrose/alignment location}{\prloc}
1822     \ifx\prloc\pgfutil@empty
1823     \else
1824     \begin{group}
1825       \tikzset{name prefix ..}%
1826       \tikz@scan@one@point\pgfutil@firstofone%
1827       (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} start)%
1828       \global\pgf@xa=\pgf@x
1829       \global\pgf@ya=\pgf@y
1830       \tikz@scan@one@point\pgfutil@firstofone%
1831       (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} end)%
1832       \global\pgf@xb=\pgf@x
1833       \global\pgf@yb=\pgf@y
1834     \end{group}
1835     \advance\pgf@xb by -\pgf@xa
1836     \advance\pgf@yb by -\pgf@ya
1837     \pgftransformshift{\pgfpoint{\pgf@xa}{\pgf@ya}}%
1838     \pgfpointnormalised{\pgfpoint{\pgf@xb}{\pgf@yb}}
1839     \pgf@xb=\pgf@x
1840     \pgf@yb=\pgf@y
1841     \pgftransformtriangle%
1842     {\pgfpoint{0pt}{0pt}}%
1843     {\pgfpoint{\pgf@xb}{\pgf@yb}}%
1844     {\pgfpoint{-\pgf@yb}{\pgf@xb}}%
1845     \test@edgealign{d}%
1846     \if@edgealign
1847     \pgftransformrotate{180}%
1848     \pgftransformshift{\pgfpoint{\pr@cphi cm - 1cm}{\pr@sphi cm}}%
1849     \else
1850     \test@edgealign{b}%
1851     \if@edgealign%
1852     \test@newedge{2}%
1853     \if@newedge
1854     \pgftransformrotate{144}%
1855     \pgftransformshift{\pgfpoint{-1 cm}{0 cm}}%
1856     \else
1857     \pgftransformrotate{216}%
1858     \pgftransformshift{\pgfpoint{\pr@cphi cm}{\pr@sphi cm}}%
1859     \fi
1860     \else
1861     \test@edgealign{c}%
1862     \if@edgealign%
1863     \test@newedge{2}%
1864     \if@newedge
1865     \pgftransformshift{\pgfpoint{1 cm}{0 cm}}%
1866     \pgftransformrotate{72}%
1867     \else

```

```

1868 \test@newedge{3}%
1869 \if@newedge
1870 \pgftransformrotate{-72}%
1871 \pgftransformshift{\pgfpoint{2 * \pr@shphi cm}{0 cm}}%
1872 \else
1873 \test@newedge{4}%
1874 \if@newedge
1875 \pgftransformshift{\pgfpoint{1 cm + 2 * \pr@shphi cm}{0 cm}}%
1876 \else
1877 \fi\fi\fi\fi\fi
1878 \fi
1879 \UsePenroseTile[
1880     every Penrose tile clip/.try,
1881     every boat clip/.try
1882 ]{boat}
1883 \UsePenroseTile[
1884     every Penrose tile/.try,
1885     every boat/.try,
1886     pic actions
1887 ]{boat}
1888 \coordinate (-edge C1 start) at (1,0);
1889 \coordinate (-edge C1 end) at (0,0);
1890 \coordinate (-edge C2 start) at (0,0);
1891 \coordinate (-edge C2 end) at (-\pr@shphi,\pr@cphi);
1892 \coordinate (-edge C3 start) at (-\pr@shphi,\pr@cphi);
1893 \coordinate (-edge C3 end) at (-2*\pr@shphi,0);
1894 \coordinate (-edge C4 start) at (-2*\pr@shphi,0);
1895 \coordinate (-edge C4 end) at (-1-2*\pr@shphi,0);
1896 \coordinate (-edge B1 start) at (-1-2*\pr@shphi,0);
1897 \coordinate (-edge B1 end) at (-\pr@cphi,-\pr@sphi);
1898 \coordinate (-edge D start) at (-\pr@cphi,-\pr@sphi);
1899 \coordinate (-edge D end) at (1-\pr@cphi,-\pr@sphi);
1900 \coordinate (-edge B2 start) at (1-\pr@cphi,-\pr@sphi);
1901 \coordinate (-edge B2 end) at (1,0);
1902 \end{scope}
1903 },
1904 boat/.style={
1905     every Penrose pic/.try,
1906     pic type=boat,
1907 },

```

The diamond.

```

1908 diamond/.pic={
1909     \begin{scope}
1910         \pgfkeysgetvalue{/tikz/Penrose/alignment location}{\prloc}
1911         \ifx\prloc\pgfutil@empty
1912             \else
1913             \begingroup
1914             \tikzset{name prefix ..}%
1915             \tikz@scan@one@point\pgfutil@firstofone%
1916             (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} start)%
1917             \global\pgf@xa=\pgf@x
1918             \global\pgf@ya=\pgf@y
1919             \tikz@scan@one@point\pgfutil@firstofone%
1920             (\prloc-edge \pgfkeysvalueof{/tikz/Penrose/alignment edge} end)%

```

```

1921 \global\pgf@xb=\pgf@x
1922 \global\pgf@yb=\pgf@y
1923 \endgroup
1924 \advance\pgf@xb by -\pgf@xa
1925 \advance\pgf@yb by -\pgf@ya
1926 \pgftransformshift{\pgfpoint{\pgf@xa}{\pgf@ya}}%
1927 \pgfpointnormalised{\pgfpoint{\pgf@xb}{\pgf@yb}}
1928 \pgf@xb=\pgf@x
1929 \pgf@yb=\pgf@y
1930 \pgftransformtriangle%
1931 {\pgfpoint{0pt}{0pt}}%
1932 {\pgfpoint{\pgf@xb}{\pgf@yb}}%
1933 {\pgfpoint{-\pgf@yb}{\pgf@xb}}%
1934 \test@edgealign{d}%
1935 \if@edgealign%
1936 \test@newedge{2}%
1937 \if@newedge
1938 \pgftransformshift{\pgfpoint{1cm}{0cm}}%
1939 \pgftransformrotate{-162}%
1940 \else
1941 \pgftransformrotate{-18}%
1942 \fi
1943 \else
1944 \test@newedge{2}%
1945 \if@newedge
1946 \pgftransformrotate{162}%
1947 \pgftransformshift{\pgfpoint{- 2 * \pr@chphi cm}{0cm}}%
1948 \else
1949 \pgftransformshift{\pgfpoint{- \pr@cphi cm}{-\pr@sphi cm}}%
1950 \pgftransformrotate{18}%
1951 \fi\fi
1952 \fi
1953 \UsePenroseTile[
1954   every Penrose tile clip/.try,
1955   every diamond clip/.try
1956 ]{diamond}
1957 \UsePenroseTile[
1958   every Penrose tile/.try,
1959   every diamond/.try,
1960   pic actions
1961 ]{diamond}
1962 \coordinate (-edge D1 start) at (0,0);
1963 \coordinate (-edge D1 end) at (\pr@chphi,\pr@shphi);
1964 \coordinate (-edge B1 start) at (\pr@chphi,\pr@shphi);
1965 \coordinate (-edge B1 end) at (2*\pr@chphi,0);
1966 \coordinate (-edge B2 start) at (2*\pr@chphi,0);
1967 \coordinate (-edge B2 end) at (\pr@chphi,-\pr@shphi);
1968 \coordinate (-edge D2 start) at (\pr@chphi,-\pr@shphi);
1969 \coordinate (-edge D2 end) at (0,0);
1970 \end{scope}
1971 },
1972 diamond/.style={
1973   every Penrose pic/.try,
1974   pic type=diamond,

```

1975 }
1976 }