

# The Tilings package: code

Andrew Stacey

[loopspace@mathforge.org](mailto:loopspace@mathforge.org)

v2.0 from 2023/06/01

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Implementation</b>	<b>1</b>
2.1	Initialisation	1
2.2	Helpful Error Messages	3
2.3	Creating the Tiles	3
2.4	Specifying the Tiles	7
2.5	Lindenmayer System	18
2.6	Lindenmayer System	24
2.7	Lindenmayer System	38

## 1 Introduction

This is a TikZ library for drawing tiles, such as Penrose tiles (kite/dart, rhombus, and pentagon versions) and the aperiodical polykite tiles. 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 full user documentation, see the `tilings.pdf` file.

## 2 Implementation

```
1 <@@=tilings>
```

### 2.1 Initialisation

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

```
2 \ProvidesExplFile
3   {tikzlibrarytilings.code.tex}
4   {2023/06/01}
5   {2.0}
6   {TikZ pics for tilings such as Penrose tiles}
7 \RequirePackage{tikz}
8 \usetikzlibrary{spath3}
```

Now we move in to the realm of L<sup>A</sup>T<sub>E</sub>X3.

```
9 \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. These are globally defined as we will frequently want to define them in one `tikzpicture` and use them in another.

```
10 \tl_new:N \g__tilings_side_a_tl
11 \tl_new:N \g__tilings_side_b_tl
12 \tl_new:N \g__tilings_side_c_tl
```

```

13 \tl_new:N \g__tilings_side_d_tl
14 \tl_new:N \g__tilings_side_e_tl
15 \tl_new:N \g__tilings_side_A_tl
16 \tl_new:N \g__tilings_side_B_tl
17 \tl_new:N \g__tilings_side_C_tl
18 \tl_new:N \g__tilings_side_D_tl
19 \tl_new:N \g__tilings_side_E_tl
20 \tl_new:c {g__tilings_side_1_tl}
21 \tl_new:c {g__tilings_side_2_tl}
22 \tl_new:c {g__tilings_side_3_tl}
23 \tl_gset:Nn \g__tilings_side_a_tl
24 {
25   \pgfsyssoftpath@movetotoken{0pt}{0pt}
26   \pgfsyssoftpath@linetotoken{1pt}{0pt}
27 }
28 \tl_gset_eq:NN \g__tilings_side_b_tl \g__tilings_side_a_tl
29 \tl_gset_eq:NN \g__tilings_side_c_tl \g__tilings_side_a_tl
30 \tl_gset_eq:NN \g__tilings_side_d_tl \g__tilings_side_a_tl
31 \tl_gset_eq:NN \g__tilings_side_e_tl \g__tilings_side_a_tl
32 \tl_gset_eq:NN \g__tilings_side_A_tl \g__tilings_side_a_tl
33 \tl_gset_eq:NN \g__tilings_side_B_tl \g__tilings_side_a_tl
34 \tl_gset_eq:NN \g__tilings_side_C_tl \g__tilings_side_a_tl
35 \tl_gset_eq:NN \g__tilings_side_D_tl \g__tilings_side_a_tl
36 \tl_gset_eq:NN \g__tilings_side_E_tl \g__tilings_side_a_tl
37 \tl_gset_eq:cN {g__tilings_side_1_tl} \g__tilings_side_a_tl
38 \tl_gset_eq:cN {g__tilings_side_2_tl} \g__tilings_side_a_tl
39 \tl_gset_eq:cN {g__tilings_side_3_tl} \g__tilings_side_a_tl

```

ngs\_tmpa\_fp We need a few temporary variables to hold intermediate calculations.

```

ngs_tmpb_fp 40 \fp_new:N \l__tilings_tmpa_fp
ngs_tmpc_fp 41 \fp_new:N \l__tilings_tmpb_fp
saved_x_fp 42 \fp_new:N \l__tilings_tmpc_fp
saved_y_fp 43 \fp_new:N \l__tilings_saved_x_fp
ngs_tmpa_str 44 \fp_new:N \l__tilings_saved_y_fp
ngs_tmpb_str 45 \str_new:N \l__tilings_tmpa_str
ngs_tmpa_seq 46 \str_new:N \l__tilings_tmpb_str
ngs_tmpa_tl 47 \seq_new:N \l__tilings_tmpa_seq
ngs_tmpb_tl 48 \tl_new:N \l__tilings_tmpa_tl
ngs_tmpc_tl 49 \tl_new:N \l__tilings_tmpb_tl
ngs_tmpd_tl 50 \tl_new:N \l__tilings_tmpc_tl
ngs_tmpa_int 51 \tl_new:N \l__tilings_tmpd_tl
ngs_tmpb_int 52 \tl_new:N \l__tilings_tmp_tile_path_tl
ngs_xa_dim 53 \tl_new:N \l__tilings_action_lms_tl
ngs_ya_dim 54 \tl_new:N \l__tilings_parameters_lms_tl
ngs_xb_dim 55 \int_new:N \l__tilings_tmpa_int
ngs_yb_dim 56 \int_new:N \l__tilings_tmpb_int
ngs_xa_dim 57 \fp_new:N \l__tilings_xa_fp
ngs_ya_dim 58 \fp_new:N \l__tilings_ya_fp
ngs_xb_dim 59 \fp_new:N \l__tilings_xb_fp
ngs_yb_dim 60 \fp_new:N \l__tilings_yb_fp
ngs_xa_dim 61 \dim_new:N \l__tilings_xa_dim
ngs_ya_dim 62 \dim_new:N \l__tilings_ya_dim
ngs_xb_dim 63 \dim_new:N \l__tilings_xb_dim
ngs_yb_dim 64 \dim_new:N \l__tilings_yb_dim
ngs_xa_dim 65 \dim_new:N \g__tilings_xa_dim
ngs_ya_dim 66 \dim_new:N \g__tilings_ya_dim
ngs_xb_dim 67 \dim_new:N \g__tilings_xb_dim
ngs_yb_dim 68 \dim_new:N \g__tilings_yb_dim
ngs_xa_dim 69 \prop_new:N \l__tilings_tmpa_prop
ngs_ya_dim 70 \bool_new:N \l__tilings_cw_bool
ngs_xb_dim 71 \bool_new:N \l__tilings_update_saved_bool

```

```

72 \bool_new:N \l__tilings_relative_bool
73 \bool_new:N \l__tilings_edge_bool
74 \str_const:Nn \c__tilings_colon_str {:}
75 \str_const:Nn \c__tilings_comma_str {,}
76 \fp_const:Nn \c__tilings_cm_fp {\dim_to_fp:n {1cm}}
77 \tl_new:N \g__tilings_output_tl
78 \fp_new:N \g__tilings_output_a_fp
79 \fp_new:N \g__tilings_output_b_fp
80 \prop_new:N \g__tilings_tilenames_prop
81 \regex_const:Nn \c__tilings_anchor_regex {\s\w+\Z}
82 \cs_generate_variant:Nn \seq_set_split:Nnn {NVV}
83 \cs_generate_variant:Nn \regex_extract_once:NnNTF {NVNTF}
84 \cs_generate_variant:Nn \tl_if_eq:nnT {nVT}
85 \cs_generate_variant:Nn \tl_if_in:NnT {NVT}
86 \cs_generate_variant:Nn \prop_item:Nn {cV}
87 \cs_generate_variant:Nn \tl_if_head_is_group_p:n {V}

```

(End definition for `\l__tilings_tmpa_fp` and others.)

## 2.2 Helpful Error Messages

```

88 \msg_new:nnn { tilings }{ not baked }
89 { Tile~ #1~ has~ not~ been~ baked. }
90 \msg_new:nnn { tilings }{ no tile }
91 { Tile~ #1~ has~ not~ been~ defined. }
92 \msg_new:nnn { tilings }{ no side }
93 { Tile~ side~ #1~ has~ not~ been~ defined,~ using~ default. }
94 \msg_new:nnn { tilings }{ tile no edge }
95 { Tile~ #1~ doesn't~ have~ an~ edge~ labelled~ #2;
96 ~ available~ edges~ are~ #3.}
97 \msg_new:nnn { tilings }{ no edge }
98 { Either~ tile~ #1~ doesn't~ exist
99 ~ or~ it~ doesn't~ have~ an~ edge~ labelled~ #2.}

```

## 2.3 Creating the Tiles

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

```

100 \cs_new_nopar:Npn \__tilings_normalise_path:Nn #1#2
101 {

```

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

```

102 \group_begin:
103 \spath_initialpoint:Nn \l__tilings_tmpa_tl {#2}
104 \fp_set:Nn \l__tilings_tmpa_fp {\tl_head:N \l__tilings_tmpa_tl}
105 \tl_set:Nx \l__tilings_tmpa_tl {\tl_tail:N \l__tilings_tmpa_tl}
106 \fp_set:Nn \l__tilings_tmpb_fp {\tl_head:N \l__tilings_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:

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

```

107 \spath_finalpoint:Nn \l__tilings_tmpa_tl {#2}
108 \fp_set:Nn \l__tilings_tmpa_fp
109 {\tl_head:N \l__tilings_tmpa_tl - \l__tilings_tmpa_fp}
110 \tl_set:Nx \l__tilings_tmpa_tl {\tl_tail:N \l__tilings_tmpa_tl}
111 \fp_set:Nn \l__tilings_tmpb_fp
112 {\tl_head:N \l__tilings_tmpa_tl - \l__tilings_tmpb_fp}

```

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

```

113 \fp_set:Nn \l__tilings_tmppc_fp
114 {\l__tilings_tmpa_fp^2 + \l__tilings_tmpb_fp^2}

```

```

115 \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_tmpa_fp/\l__tilings_tmpe_fp}
116 \fp_set:Nn \l__tilings_tmpe_fp {\l__tilings_tmpe_fp/\l__tilings_tmpe_fp}
117 \fp_set:Nn \l__tilings_tmpe_fp {-\l__tilings_tmpe_fp}

```

Now construct the matrix.

```

118 \tl_set:Nx \l__tilings_tmpe_tl
119 {
120   {\fp_use:N \l__tilings_tmpa_fp}
121   {\fp_use:N \l__tilings_tmpe_fp} % swapped
122   {\fp_use:N \l__tilings_tmpe_fp} % swapped
123   {\fp_use:N \l__tilings_tmpa_fp}
124 }

```

Get the initial point back again for the translation part.

```

125 \spath_initialpoint:Nn \l__tilings_tmpe_tl {#2}

```

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

```

126 \fp_set:Nn \l__tilings_tmpe_fp
127 {
128   (-1) * \l__tilings_tmpe_fp * \tl_head:N \l__tilings_tmpe_tl
129   + (-1) * \l__tilings_tmpe_fp * \tl_tail:N \l__tilings_tmpe_tl
130 }
131 \fp_set:Nn \l__tilings_tmpe_fp
132 {
133   (-1) * \l__tilings_tmpe_fp * \tl_tail:N \l__tilings_tmpe_tl
134   + \l__tilings_tmpe_fp * \tl_head:N \l__tilings_tmpe_tl
135 }

```

Finally, we apply the transformation to the path.

```

136 \tl_put_right:Nx \l__tilings_tmpe_tl {
137   {\fp_to_dim:N \l__tilings_tmpe_fp}
138   {\fp_to_dim:N \l__tilings_tmpe_fp}
139 }
140 \spath_transform:NnV \l__tilings_tmpe_tl {#2} \l__tilings_tmpe_tl
141 \tl_gset_eq:NN \g__tilings_output_tl \l__tilings_tmpe_tl
142 \group_end:
143 \tl_set_eq:NN #1 \g__tilings_output_tl
144 \tl_gclear:N \g__tilings_output_tl
145 }
146 \cs_generate_variant:Nn \__tilings_normalise_path:Nn {NV, cn, cV}
147 \cs_new_protected_nopar:Npn \__tilings_normalise_path:N #1
148 {
149   \__tilings_normalise_path:NV #1#1
150 }
151 \cs_generate_variant:Nn \__tilings_normalise_path:N {c}

```

*(End definition for \\_\_tilings\_normalise\_path:Nn.)*

---

**SetTilingPath** This sets the path corresponding to a particular side to the current path, and normalises it.

```

152 \cs_new_protected_nopar:Npn \__tilings_set_tiling_path:n #1
153 {
154   \group_begin:
155   \pgfsyssoftpath@getcurrentpath\l__tilings_tmpe_tl
156   \__tilings_normalise_path:N \l__tilings_tmpe_tl
157   \tl_gset_eq:cN {g__tilings_side_#1_tl} \l__tilings_tmpe_tl
158   \group_end:
159 }
160
161 \NewDocumentCommand \SetTilingPath { m }
162 {
163   \__tilings_set_tiling_path:n {#1}
164 }

```

`\tikz@scan@one@point:n` This is a wrapper around `\tikz@scan@one@point` to make it easier to use with L<sup>A</sup>T<sub>E</sub>X3 variables.

```
165 \cs_new_nopar:Npn \tikz_scan_point:n #1
166 {
167   \tikz@scan@one@point\pgfutil@firstofone#1\relax
168 }
169 \cs_generate_variant:Nn \tikz_scan_point:n {V}
```

*(End definition for \tikz\_scan\_point:n.)*

`\tikz_node_if_defined:TF` This extracts the code that tests if a node is defined.

```
170 \prg_new_conditional:Npnn \tikz_node_if_defined:n #1 {p,T,F,TF}
171 {
172   \tl_if_exist:cTF {pgf@sh@ns@\use:c{tikz@pp@name}{#1}}
173   {
174     \prg_return_true:
175   }{
176     \tl_if_exist:cTF
177     {pgf@sh@ns@not yet positionedPGFINTERNAL\use:c{tikz@pp@name}{#1}}
178     {
179       \pgf_return_true:
180     }
181     {
182       \prg_return_false:
183     }
184   }
185 }
```

*(End definition for \tikz\_node\_if\_defined:TF.)*

`\pgfkeys_get:Nn` This is a wrapper around `\pgfkeysgetvalue` to make it easier to use with L<sup>A</sup>T<sub>E</sub>X3 variables.

```
\pgfkeys_get:n
186 \cs_new_nopar:Npn \__tilings_keys_get:Nn #1#2
187 {
188   \pgfkeysgetvalue{/tikz/tiling/#2}{#1}
189 }
190 \cs_new_nopar:Npn \__tilings_keys_get:n #1
191 {
192   \pgfkeysvalueof{/tikz/tiling/#1}
193 }
194 \cs_new_nopar:Npn \__tilings_tikz_keys_get:Nn #1#2
195 {
196   \pgfkeysgetvalue{/tikz/#2}{#1}
197 }
198 \cs_new_nopar:Npn \__tilings_tikz_keys_get:n #1
199 {
200   \pgfkeysvalueof{/tikz/#1}
201 }
202 \cs_new_nopar:Npn \__tilings_pgf_keys_get:Nn #1#2
203 {
204   \pgfkeysgetvalue{#2}{#1}
205 }
206 \cs_new_nopar:Npn \__tilings_pgf_keys_get:n #1
207 {
208   \pgfkeysvalueof{#1}
209 }
```

*(End definition for \\_\_tilings\_keys\_get:Nn and \pgfkeys\_get:n.)*

`\ke_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.

```
210 \cs_new_nopar:Npn \__tilings_make_tile:nnn #1#2#3
211 {
```

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

```

212 \group_begin:
213 \tl_set:Nn \l__tilings_tmpa_tl {#3}
214 \tl_set:Nx \l__tilings_tmpb_tl {\tl_head:N \l__tilings_tmpa_tl}
215 \tl_set:Nn \l__tilings_tmpa_tl {\pgfsyssoftpath@movetotoken}
216 \tl_put_right:Nx \l__tilings_tmpa_tl
217 {
218   {
219     \fp_to_dim:n
220     {(\tl_item:Nn \l__tilings_tmpb_tl {1}) * \c__tilings_cm_fp}
221   }
222   {
223     \fp_to_dim:n
224     {(\tl_item:Nn \l__tilings_tmpb_tl {2}) * \c__tilings_cm_fp}
225   }
226 }
227 \tl_set_eq:NN \l__tilings_tmp_tile_path_tl \l__tilings_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.

```

228 \tl_set:Nn \l__tilings_tmpa_tl {#3}
229 \tl_put_right:Nx \l__tilings_tmpa_tl {\tl_head:N \l__tilings_tmpa_tl}

```

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

```

230 \tl_map_inline:nn {#2} {

```

Clone the path for this side.

```

231 \tl_set:Nx \l__tilings_tmpe_tl {\tl_head:n {##1}}
232 \tl_if_exist:cF {g__tilings_side_ \tl_use:N \l__tilings_tmpe_tl _tl}
233 {
234   \msg_error:nxx { tilings }{ no side } { \tl_use:N \l__tilings_tmpe_tl }
235   \tl_gset_eq:cc
236   {g__tilings_side_ \tl_use:N \l__tilings_tmpe_tl _tl}
237   {g__tilings_side_a_tl}
238 }
239 \tl_set_eq:Nc \l__tilings_tmpe_tl
240 {g__tilings_side_ \tl_use:N \l__tilings_tmpe_tl _tl}

```

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

```

241 \tl_set:Nx \l__tilings_tmpe_tl {\tl_head:N \l__tilings_tmpa_tl}
242 \tl_set:Nx \l__tilings_tmpe_tl {\tl_tail:N \l__tilings_tmpa_tl}

```

Store the resulting coordinate.

```

243 \fp_set:Nn \l__tilings_tmpe_fp
244 { \tl_item:Nn \l__tilings_tmpe_tl {1} }
245 \fp_set:Nn \l__tilings_tmpe_fb
246 { \tl_item:Nn \l__tilings_tmpe_tl {2} }

```

Now get the next coordinate.

```

247 \tl_set:Nx \l__tilings_tmpe_tl {\tl_head:N \l__tilings_tmpa_tl}

```

We want the difference between the two coordinates.

```

248 \fp_set:Nn \l__tilings_tmpe_fa
249 {\tl_item:Nn \l__tilings_tmpe_tl {1} - \l__tilings_tmpe_fb}
250 \fp_set:Nn \l__tilings_tmpe_fb
251 {\tl_item:Nn \l__tilings_tmpe_tl {2} - \l__tilings_tmpe_fb}

```

This is converted into a transformation matrix.

```

252 \fp_set:Nn \l__tilings_tmpe_fm {-\l__tilings_tmpe_fb}
253 \tl_set:Nx \l__tilings_tmpe_fm
254 {
255   {\fp_to_dim:n { \l__tilings_tmpe_fa * \c__tilings_cm_fm }}
256   {\fp_to_dim:n { \l__tilings_tmpe_fb * \c__tilings_cm_fm }}% not swapped

```

```

257     {\fp_to_dim:n { \l__tilings_tmpc_fp * \c__tilings_cm_fp }}% not swapped
258     {\fp_to_dim:n { \l__tilings_tmpa_fp * \c__tilings_cm_fp }}
259     {0}
260     {0}
261   }

```

The transformation is applied to the cloned path.

```

262   \spath_transform:NV \l__tilings_tmpd_tl \l__tilings_tmpb_tl

```

And this is welded to the tile path.

```

263   \spath_weld:NV \l__tilings_tmp_tile_path_tl \l__tilings_tmpd_tl
264 }

```

At the end we close the path.

```

265 \spath_close:N \l__tilings_tmp_tile_path_tl
266 \tl_gset_eq:NN \g__tilings_output_tl \l__tilings_tmp_tile_path_tl
267 \group_end:
268 \tl_gclear_new:c {g__tilings_tile_#1_tl}
269 \tl_gset_eq:cN {g__tilings_tile_#1_tl} \g__tilings_output_tl
270 \tl_gclear:N \g__tilings_output_tl
271 }

```

*(End definition for \\_\_tilings\_make\_tile:nnn.)*

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

```

272 \cs_new_nopar:Npn \__tilings_make_tile:nn #1#2
273 {
274   \__tilings_make_tile:nnn {#1} #2
275 }
276 \cs_generate_variant:Nn \__tilings_make_tile:nn {nV}

```

*(End definition for \\_\_tilings\_make\_tile:nn.)*

## 2.4 Specifying the Tiles

The tile specifications are contained in a prop.

```

277 \prop_new:N \g__tilings_tiles_prop

```

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

`ordinate:w`

```

278 \cs_new_nopar:Npn \__tilings_add_coordinate:Nnn #1#2#3 {
279   \group_begin:
280   \fp_set:Nn \l__tilings_tmpa_fp {#2}
281   \fp_set:Nn \l__tilings_tmpb_fp {#3}
282   \bool_if:NT \l__tilings_relative_bool
283   {
284     \fp_add:Nn \l__tilings_tmpa_fp {\l__tilings_saved_x_fp}
285     \fp_add:Nn \l__tilings_tmpb_fp {\l__tilings_saved_y_fp}
286   }
287   \fp_gset_eq:NN \g__tilings_output_a_fp \l__tilings_tmpa_fp
288   \fp_gset_eq:NN \g__tilings_output_b_fp \l__tilings_tmpb_fp
289   \group_end:
290   \tl_put_right:Nx #1
291   {
292     {\fp_use:N \g__tilings_output_a_fp}{\fp_use:N \g__tilings_output_b_fp}}
293   }
294   \bool_if:NT \l__tilings_update_saved_bool
295   {
296     \fp_set_eq:NN \l__tilings_saved_x_fp \g__tilings_output_a_fp
297     \fp_set_eq:NN \l__tilings_saved_y_fp \g__tilings_output_b_fp
298   }
299   \fp_gzero:N \g__tilings_output_a_fp
300   \fp_gzero:N \g__tilings_output_b_fp
301 }

```

Wrapper around the add coordinate command to split at a comma.

```

302 \cs_new_nopar:Npn \__tilings_add_xy_coordinate:w #1#2,#3 \q_stop
303 {
304   \__tilings_add_coordinate:Nnn #1 {#2}{#3}
305 }

```

Wrapper around the add coordinate command to split at a colon.

```

306 \cs_new_nopar:Npn \__tilings_add_rth_coordinate:w #1#2:#3 \q_stop
307 {
308   \__tilings_add_coordinate:Nnn #1 {(#3) * cosd(#2)}{(#3) * sind(#2)}
309 }

```

(End definition for `\__tilings_add_coordinate:Nnn` and `\__tilings_add_coordinate:w`.)

side\_to\_axis:Nnn Apply a transformation to make a given side lie on the x-axis. Second argument is the tile, third is the side, first is whether to reverse the side.

```

310 \cs_new_nopar:Npn \__tilings_transform_side_to_axis:Nnn #1#2#3
311 {

```

Get our tile data, checking if the tile exists.

```

312   \group_begin:
313   \prop_get:NnNTF \g__tilings_tiles_prop {#2} \l__tilings_tmpa_tl
314   {

```

Start with the edge list.

Initialise the counter.

```

315   \int_zero:N \l__tilings_tmpb_int
316   \int_incr:N \l__tilings_tmpb_int

```

Get the path type list.

```

317   \tl_set:Nx \l__tilings_tmpc_tl {\tl_head:N \l__tilings_tmpa_tl}

```

Iterate through the path type list, looking for the requested path.

```

318   \bool_set_false:N \l__tilings_edge_bool
319   \tl_map_inline:Nn \l__tilings_tmpc_tl {
320     \str_if_eq:nnT {##1} {#3} {
321       \bool_set_true:N \l__tilings_edge_bool
322       \tl_map_break:
323     }
324     \int_incr:N \l__tilings_tmpb_int
325   }
326   \bool_if:NTF \l__tilings_edge_bool
327   {

```

Get the coordinate list.

```

328     \tl_set:Nx \l__tilings_tmpc_tl {\tl_tail:N \l__tilings_tmpa_tl}

```

Strip off the outer braces.

```

329     \tl_set:Nx \l__tilings_tmpc_tl {\tl_item:Nn \l__tilings_tmpc_tl {1}}

```

Add the first coordinate at the end.

```

330     \tl_put_right:Nx \l__tilings_tmpc_tl
331     {{\tl_item:Nn \l__tilings_tmpc_tl {1}}}

```

Get the coordinates for this edge.

```

332     \tl_set:Nx \l__tilings_tmpa_tl
333     {\tl_item:Nn \l__tilings_tmpc_tl {\int_use:N \l__tilings_tmpb_int}}
334     \tl_set:Nx \l__tilings_tmpb_tl
335     {\tl_item:Nn \l__tilings_tmpc_tl {\int_use:N \l__tilings_tmpb_int + 1}}

```



Possibly swap the ends.

```

336     \bool_if:NT #1
337     {
338         \tl_set:NV \l__tilings_tmpc_tl \l__tilings_tmpa_tl
339         \tl_set:NV \l__tilings_tmpa_tl \l__tilings_tmpb_tl
340         \tl_set:NV \l__tilings_tmpb_tl \l__tilings_tmpc_tl
341     }

```

Get the coordinates of the first point, which will be the origin of the transformation.

```

342     \fp_set:Nn \l__tilings_xa_fp {\tl_item:Nn \l__tilings_tmpb_tl {1}}
343     \fp_set:Nn \l__tilings_ya_fp {\tl_item:Nn \l__tilings_tmpb_tl {2}}

```

Get the coordinates of the second point and adjust relative to the first.

```

344     \fp_set:Nn \l__tilings_xb_fp
345     {\tl_item:Nn \l__tilings_tmpa_tl {1} - \l__tilings_xa_fp}
346     \fp_set:Nn \l__tilings_yb_fp
347     {\tl_item:Nn \l__tilings_tmpa_tl {2} - \l__tilings_ya_fp}

```

And normalise the vector along it.

```

348 % \fp_set:Nn \l__tilings_xb_fp {\l__tilings_xb_fp / \c__tilings_cm_fp}
349 % \fp_set:Nn \l__tilings_yb_fp {\l__tilings_yb_fp / \c__tilings_cm_fp}
350     \fp_set:Nn \l__tilings_tmpa_fp
351     {(\l__tilings_xb_fp)^2 + (\l__tilings_yb_fp)^2}
352     \fp_set:Nn \l__tilings_xb_fp { \l__tilings_xb_fp / \l__tilings_tmpa_fp}
353     \fp_set:Nn \l__tilings_yb_fp { \l__tilings_yb_fp / \l__tilings_tmpa_fp}

```

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

```

354     \tl_gset:Nx \g__tilings_output_tl
355     {
356         \exp_not:N \pgftransformtriangle
357         {
358             \exp_not:N \pgfpoint{Opt}{Opt}
359         }
360         {
361             \exp_not:N \pgfpoint
362             {\fp_to_dim:N \l__tilings_xb_fp}{\fp_to_dim:n {-\l__tilings_yb_fp}}
363         }
364         {
365             \exp_not:N \pgfpoint
366             {\fp_to_dim:N \l__tilings_yb_fp}{\fp_to_dim:N \l__tilings_xb_fp}
367         }
368         \exp_not:N \pgftransformshift
369         {
370             \exp_not:N \pgfpoint
371             {
372                 \fp_to_dim:n {-\l__tilings_xa_fp * \c__tilings_cm_fp}
373             }
374             {
375                 \fp_to_dim:n {-\l__tilings_ya_fp * \c__tilings_cm_fp}
376             }
377         }
378     }
379 }
380 {
381     \msg_error:nnxxx {tilings} {tile no edge} {#2} {#3}
382     {\tl_use:N \l__tilings_tmpc_tl }
383     \tl_gclear:N \g__tilings_output_tl
384 }
385 }
386 {
387     \msg_error:nnn {tilings} {no tile} {#2}
388     \tl_gclear:N \g__tilings_output_tl

```

```

389 }
390 \group_end:
391 \tl_use:N \g__tilings_output_tl
392 \tl_gclear:N \g__tilings_output_tl
393 }

```

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

```

394 \cs_generate_variant:Nn
395 \__tilings_transform_side_to_axis:Nnn {Nnx,NnV,NVV}

```

`ex_to_origin:Nnn` Apply a transformation to make a given vertex sit at the origin. Second argument is the tile, third is the side, first is a boolean to determine whether to use the start or end.

```

396 \cs_new_nopar:Npn \__tilings_translate_vertex_to_origin:Nnn #1#2#3
397 {

```

Get our tile data, checking if the tile exists.

```

398 \group_begin:
399 \prop_get:NnNTF \g__tilings_tiles_prop {#2} \l__tilings_tmpa_tl
400 {

```

Start with the edge list.

Initialise the counter.

```

401 \int_zero:N \l__tilings_tmpb_int
402 \int_incr:N \l__tilings_tmpb_int

```

Get the path type list.

```

403 \tl_set:Nx \l__tilings_tmpc_tl {\tl_head:N \l__tilings_tmpa_tl}

```

Iterate through the path type list, looking for the requested path.

```

404 \bool_set_false:N \l__tilings_edge_bool
405 \tl_map_inline:Nn \l__tilings_tmpc_tl {
406   \str_if_eq:nnT {##1} {#3} {
407     \bool_set_true:N \l__tilings_edge_bool
408     \tl_map_break:
409   }
410   \int_incr:N \l__tilings_tmpb_int
411 }
412 \bool_if:NTF \l__tilings_edge_bool
413 {

```

Get the coordinate list.

```

414 \tl_set:Nx \l__tilings_tmpc_tl {\tl_tail:N \l__tilings_tmpa_tl}

```

Strip off the outer braces.

```

415 \tl_set:Nx \l__tilings_tmpc_tl {\tl_item:Nn \l__tilings_tmpc_tl {1}}

```

Add the first coordinate at the end.

```

416 \tl_put_right:Nx \l__tilings_tmpc_tl
417 {{\tl_item:Nn \l__tilings_tmpc_tl {1}}}

```

Get the coordinates for this edge.

```

418 \tl_set:Nx \l__tilings_tmpa_tl
419 {\tl_item:Nn \l__tilings_tmpc_tl {\int_use:N \l__tilings_tmpb_int}}
420 \tl_set:Nx \l__tilings_tmpb_tl
421 {\tl_item:Nn \l__tilings_tmpc_tl {\int_use:N \l__tilings_tmpb_int + 1}}

```

Possibly swap the ends.

```

422 \bool_if:NT #1
423 {
424   \tl_set:NV \l__tilings_tmpc_tl \l__tilings_tmpa_tl
425   \tl_set:NV \l__tilings_tmpa_tl \l__tilings_tmpb_tl
426   \tl_set:NV \l__tilings_tmpb_tl \l__tilings_tmpc_tl
427 }

```

Get the coordinates of the first point, which will be the origin of the transformation.

```
428 \fp_set:Nn \l__tilings_xa_fp {\tl_item:Nn \l__tilings_tmpb_tl {1}}
429 \fp_set:Nn \l__tilings_ya_fp {\tl_item:Nn \l__tilings_tmpb_tl {2}}
```

Shift to place the selected vertex at the origin.

```
430 \tl_gset:Nx \g__tilings_output_tl
431 {
432 \exp_not:N \pgftransformshift
433 {
434 \exp_not:N \pgfpoint
435 {
436 \fp_to_dim:n {-\l__tilings_xa_fp * \c__tilings_cm_fp}
437 }
438 {
439 \fp_to_dim:n {-\l__tilings_ya_fp * \c__tilings_cm_fp}
440 }
441 }
442 }
443 }
444 {
445 \msg_error:nnxxx {tilings} {tile no edge} {#2} {#3}
446 {\tl_use:N \l__tilings_tmpe_tl }
447 \tl_gclear:N \g__tilings_output_tl
448 }
449 }
450 {
451 \msg_error:nnn {tilings} {no tile} {#2}
452 \tl_gclear:N \g__tilings_output_tl
453 }
454 \group_end:
455 \tl_use:N \g__tilings_output_tl
456 \tl_gclear:N \g__tilings_output_tl
457 }
```

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

```
458 \cs_generate_variant:Nn
459 \__tilings_translate_vertex_to_origin:Nnn {Nnx,NnV,NVV}
```

`cmAlongSide` Make this available outside the L<sup>A</sup>T<sub>E</sub>X<sub>3</sub> environment. The starred version allows for reversing the side.

```
460 \DeclareDocumentCommand \TransformAlongSide {s m m}
461 {
```

Store the star

```
462 \IfBooleanTF {#1}
463 {
464 \bool_set_true:N \l__tilings_cw_bool
465 }
466 {
467 \bool_set_false:N \l__tilings_cw_bool
468 }
469 \__tilings_transform_side_to_axis:Nnx \l__tilings_cw_bool {#2}{#3}
470 }
```

(End definition for `\TransformAlongSide`.)

`es_at_vertices:n` This places TikZ coordinates at the vertices of the tile.

```
471 \cs_new_nopar:Npn \__tilings_coordinates_at_vertices:n #1
472 {
473 \group_begin:
```

Get our tile data

```
474 \prop_get:NnN \g__tilings_tiles_prop {#1} \l__tilings_tmpa_tl
```

Start with the edge list

```
475 \tl_set:Nx \l__tilings_tmpb_tl {\tl_head:N \l__tilings_tmpa_tl}
```

Get the coordinate list

```
476 \tl_set:Nx \l__tilings_tmpc_tl {\tl_tail:N \l__tilings_tmpa_tl}
```

Strip off the outer braces

```
477 \tl_set:Nx \l__tilings_tmpc_tl
478 {\tl_item:Nn \l__tilings_tmpc_tl {1}}
```

Add the first coordinate at the end

```
479 \tl_put_right:Nx \l__tilings_tmpc_tl
480 {\tl_item:Nn \l__tilings_tmpc_tl {1}}
```

Get the first coordinate

```
481 \tl_set:Nx \l__tilings_tmpa_tl {\tl_head:N \l__tilings_tmpc_tl}
482 \tl_set:Nx \l__tilings_tmpc_tl {\tl_tail:N \l__tilings_tmpc_tl}
```

Iterate through the edge list, placing coordinates

```
483 \tl_map_inline:Nn \l__tilings_tmpb_tl {
484   \tl_set:Nx \l__tilings_tmpe_tl {
485     \exp_not:N \coordinate
486     (-edge~ ##1~ start)~
487     at (
488       \tl_item:Nn \l__tilings_tmpa_tl {1},
489       \tl_item:Nn \l__tilings_tmpa_tl {2}
490     );
491   }
492   \tl_use:N \l__tilings_tmpe_tl
493   \tl_set:Nx \l__tilings_tmpa_tl {\tl_head:N \l__tilings_tmpe_tl}
494   \tl_set:Nx \l__tilings_tmpe_tl {\tl_tail:N \l__tilings_tmpe_tl}
495   \tl_set:Nx \l__tilings_tmpe_tl {
496     \exp_not:N \coordinate
497     (-edge~ ##1~ end)~
498     at (
499       \tl_item:Nn \l__tilings_tmpa_tl {1},
500       \tl_item:Nn \l__tilings_tmpa_tl {2}
501     );
502   }
503   \tl_use:N \l__tilings_tmpe_tl
504 }
505 \group_end:
506 }
```

*(End definition for \\_\_tilings\_coordinates\_at\_vertices:n.)*

CoordinatesAtVertices User-accessible wrapper around the above.

```
507 \DeclareDocumentCommand \CoordinatesAtVertices {m}
508 {
509   \__tilings_coordinates_at_vertices:n {#1}
510 }
```

*(End definition for \CoordinatesAtVertices.)*

```
511 \tikzset{
512   transform~ to~ tile/.code~ args={#1~ along~ #2}{%
513     \group_begin:
514     \tl_if_in:nnTF {#1} {back}
515     {
516       \tikzset{
517         tiling/alignment~ set~ location=#1,
518         tiling/alignment~ direction={backwards}
519       }
520     }
```

```

521 {
522   \tikzset{
523     tiling/alignment~ location=#1,
524     tiling/alignment~ direction={forewards}
525   }
526 }
527 \tl_if_in:nnTF {#2} {using}
528 {
529   \tikzset{
530     tiling/alignment~ set~ edges=#2,
531   }
532 }
533 {
534   \tikzset{
535     tiling/alignment~ edge=#2,
536   }
537 }
538 \tikz_scan_point:n {
539   (\__tilings_keys_get:n {alignment~ location}
540   -edge~ \__tilings_keys_get:n {alignment~ edge}~ start)
541 }
542 \dim_set_eq:Nc \l__tilings_xa_dim {pgf@x}
543 \dim_set_eq:Nc \l__tilings_ya_dim {pgf@y}
544 \tikz_scan_point:n {
545   (\__tilings_keys_get:n {alignment~ location}
546   -edge~ \__tilings_keys_get:n {alignment~ edge}~ end)
547 }
548 \dim_set_eq:Nc \l__tilings_xb_dim {pgf@x}
549 \dim_set_eq:Nc \l__tilings_yb_dim {pgf@y}
550 \__tilings_keys_get:Nn \l__tilings_tmpb_tl {alignment~ direction}
551 \tl_if_eq:NnTF \l__tilings_tmpb_tl {forewards}
552 {
553   \dim_gset_eq:NN \g__tilings_xa_dim \l__tilings_xa_dim
554   \dim_gset_eq:NN \g__tilings_ya_dim \l__tilings_ya_dim
555   \dim_gset_eq:NN \g__tilings_xb_dim \l__tilings_xb_dim
556   \dim_gset_eq:NN \g__tilings_yb_dim \l__tilings_yb_dim
557 }
558 {
559   \dim_gset_eq:NN \g__tilings_xa_dim \l__tilings_xb_dim
560   \dim_gset_eq:NN \g__tilings_ya_dim \l__tilings_yb_dim
561   \dim_gset_eq:NN \g__tilings_xb_dim \l__tilings_xa_dim
562   \dim_gset_eq:NN \g__tilings_yb_dim \l__tilings_ya_dim
563 }
564 \dim_gsub:Nn \g__tilings_xb_dim {\g__tilings_xa_dim}
565 \dim_gsub:Nn \g__tilings_yb_dim {\g__tilings_ya_dim}
566 \dim_gset:Nn \g__tilings_xb_dim
567 {\g__tilings_xb_dim * \dim_ratio:nn {1pt}{1cm}}
568 \dim_gset:Nn \g__tilings_yb_dim
569 {\g__tilings_yb_dim * \dim_ratio:nn {1pt}{1cm}}
570 \group_end:

```

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.

We shift to the start of the edge.

```

571 \pgftransformshift{\pgfpoint{\g__tilings_xa_dim}{\g__tilings_ya_dim}}

```

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

```

572 \pgftransformtriangle
573 {\pgfpoint{0pt}{0pt}}
574 {\pgfpoint{\g__tilings_xb_dim}{\g__tilings_yb_dim}}
575 {\pgfpoint{-\g__tilings_yb_dim}{\g__tilings_xb_dim}}

```

```

576 },
577 align~ with/.code~ args={#1~ along~ #2}{%
578   \tl_if_in:nnTF {#1} {back}
579   {
580     \tikzset{
581       tiling/alignment~ set~ location=#1,
582       tiling/alignment~ direction={backwards}
583     }
584   }
585   {
586     \tikzset{
587       tiling/alignment~ location=#1,
588       tiling/alignment~ direction={forwards}
589     }
590   }
591   \tl_if_in:nnTF {#2} {using}
592   {
593     \tikzset{
594       tiling/alignment~ set~ edges=#2,
595     }
596   }
597   {
598     \tikzset{
599       tiling/alignment~ edge=#2,
600     }
601   }
602   \tikz_node_if_defined:nTF
603   {
604     \__tilings_keys_get:n {alignment~ location}
605     -edge~ \__tilings_keys_get:n {alignment~ edge}~ start
606   }
607   {
608     \tikzset{
609       tiling/alignment~ start/.expanded={
610         (\__tilings_keys_get:n {alignment~ location}
611         -edge~ \__tilings_keys_get:n {alignment~ edge}~ start)
612       },
613       tiling/alignment~ end/.expanded={
614         (\__tilings_keys_get:n {alignment~ location}
615         -edge~ \__tilings_keys_get:n {alignment~ edge}~ end)
616       },
617     }
618   }
619   {
620     \__tilings_keys_get:Nn \l__tilings_tmpa_tl {alignment~ location}
621     \tl_set:Nx \l__tilings_tmpa_tl {\tl_use:N \l__tilings_tmpa_tl}
622     \prop_get:NVNTF \g__tilings_tilenames_prop
623     \l__tilings_tmpa_tl \l__tilings_tmpb_tl
624     {
625       \prop_get:NVN \g__tilings_tiles_prop
626       \l__tilings_tmpb_tl \l__tilings_tmpc_tl
627       \msg_error:nnxxx { tilings }{ tile no edge }
628       {
629         \tl_use:N \l__tilings_tmpa_tl \c_space_tl
630         (type~ \tl_use:N \l__tilings_tmpb_tl)
631       }
632       {\__tilings_keys_get:n {alignment~ edge} }
633       { \tl_item:Nn \l__tilings_tmpc_tl {1} }
634     }
635     {
636       \msg_error:nnx { tilings }{ no tile }

```

```

637     {\_tilings_keys_get:n {alignment~ location} }
638   }
639 }
640 },
641 tiling/.is~ family,
642 tiling/alignment~ set~ location/.code~ args={#1~ back}{
643   \tikzset{
644     tiling/alignment~ location=#1,
645   }
646 },
647 tiling/alignment~ set~ edges/.code~ args={#1~ using~ #2}{
648   \tikzset{
649     tiling/alignment~ edge=#1,
650     tiling/alignment~ new~ edge=#2
651   }
652 },
653 align~ between/.code~ args={#1~ and~ #2~ using~ #3}{
654   \tikzset{
655     tiling/alignment~ start={#1},
656     tiling/alignment~ end={#2},
657   }
658   \str_set:Nn \l__tilings_tmpa_str {#3}
659   \str_set:Nx \l__tilings_tmpb_str {\str_tail:N \l__tilings_tmpa_str}
660   \tikzset{
661     tiling/alignment~ new~ edge/.expanded={\str_use:N \l__tilings_tmpb_str}
662   }
663   \str_set:Nx \l__tilings_tmpa_str {\str_head:N \l__tilings_tmpa_str}
664   \str_set:Nx \l__tilings_tmpb_str {\str_lowercase:f { \l__tilings_tmpa_str}}
665   \str_if_eq:NNT \l__tilings_tmpa_str \l__tilings_tmpb_str
666   {
667     \str_set:Nx \l__tilings_tmpb_str
668     {\str_uppercase:f { \l__tilings_tmpa_str}}
669   }
670   \tikzset{
671     tiling/alignment~ edge/.expanded={\str_use:N \l__tilings_tmpb_str},
672   }
673 },
674 tiling/alignment~ location/.initial={},
675 tiling/alignment~ edge/.initial=a,
676 tiling/alignment~ new~ edge/.initial={},
677 tiling/alignment~ direction/.initial={forewards},
678 tiling/alignment~ start/.initial={},
679 tiling/alignment~ end/.initial={},
680 tiling/anchor/.initial={},

```

Default clipping style.

```

681 every~ tile~ clip/.style={clip}
682 }

```

\DefineTile This is the user function for defining a tile.

```
683 \DeclareDocumentCommand \DefineTile { s m m m }
684 {
```

Clear the temporary variable.

```
685 \tl_clear:N \l__tilings_tmpa_tl
```

The 3rd parameter is a list of coordinates at vertices, iterate through them and add them to the list.

```
686 \int_zero:N \l__tilings_tmpa_int
687 \fp_zero:N \l__tilings_saved_x_fp
688 \fp_zero:N \l__tilings_saved_y_fp
689 \tl_map_inline:nn {#4} {
690   \str_set:Nn \l__tilings_tmpa_str {##1}
691   \str_if_eq:VnTF \l__tilings_tmpa_str {+}
692   {
693     \int_incr:N \l__tilings_tmpa_int
694   }
695   {
696     \int_case:nn {\l__tilings_tmpa_int}
697     {
698       {0} {
699         \bool_set_false:N \l__tilings_relative_bool
700         \bool_set_true:N \l__tilings_update_saved_bool
701       }
702       {1} {
703         \bool_set_true:N \l__tilings_relative_bool
704         \bool_set_false:N \l__tilings_update_saved_bool
705       }
706       {2} {
707         \bool_set_true:N \l__tilings_relative_bool
708         \bool_set_true:N \l__tilings_update_saved_bool
709       }
710     }
711   }
712   \str_if_in:NnTF \l__tilings_tmpa_str {:}
713   {
714     \seq_set_split:NVV \l__tilings_tmpa_seq \c__tilings_colon_str \l__tilings_tmpa_str
715     \__tilings_add_coordinate:Nnn \l__tilings_tmpa_tl
716     {
717       (\seq_item:Nn \l__tilings_tmpa_seq {2}) * cosd (\seq_item:Nn \l__tilings_tmpa_seq {1})
718     }
719     {
720       (\seq_item:Nn \l__tilings_tmpa_seq {2}) * sind (\seq_item:Nn \l__tilings_tmpa_seq {1})
721     }
722   }
723   {
724     \seq_set_split:NVV \l__tilings_tmpa_seq \c__tilings_comma_str \l__tilings_tmpa_str
725     \__tilings_add_coordinate:Nnn \l__tilings_tmpa_tl
726     {
727       (\seq_item:Nn \l__tilings_tmpa_seq {1})
728     }
729     {
730       (\seq_item:Nn \l__tilings_tmpa_seq {2})
731     }
732   }
733   \int_zero:N \l__tilings_tmpa_int
734 }
735 }
```

Now we make a list of the edge types (from the 2nd parameter), using a prop to keep track of whether an edge is repeated.

```
736 \prop_clear:N \l__tilings_tmpa_prop
737 \tl_map_inline:nn {#3} {
738   \prop_if_in:NnTF \l__tilings_tmpa_prop {##1}
739   {
740     \prop_put:Nnn \l__tilings_tmpa_prop {##1} {1}
741   }
742   {
743     \prop_put:Nnn \l__tilings_tmpa_prop {##1} {0}
744   }
}
```



---

**\BakeTile** This is the user wrapper around the tile creation macros.

```
903 \cs_new_protected_nopar:Npn \__tilings_bake_tile:n #1
904 {
905   \prop_get:NnN \g__tilings_tiles_prop {#1} \l__tilings_tmpa_tl
906   \__tilings_make_tile:nV {#1} \l__tilings_tmpa_tl
907 }
908
909 \NewDocumentCommand \BakeTile {m}
910 {
911   \__tilings_bake_tile:n {#1}
912 }
```

---

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

```
913 \cs_new_protected_nopar:Npn \__tilings_use_tile:nn #1#2
914 {
```

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.

```
915   \tl_if_exist:cTF {g__tilings_tile_#2_tl}
916   {
917     \tl_set_eq:Nc \l__tilings_tmp_tile_path_tl {g__tilings_tile_#2_tl}
```

We get the current transformation to apply to this path.

```
918     \pgfgettransform \l__tilings_tmpa_tl
```

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

```
919     \spath_transform:NV \l__tilings_tmp_tile_path_tl \l__tilings_tmpa_tl
920     \spath_tikz_path:nV {#1} \l__tilings_tmp_tile_path_tl
921   }
922   {
923     \msg_error:nnn { tilings } { not baked } {#2}
924   }
925 }
926
927 \NewDocumentCommand \UseTile {0{ } m}
928 {
929   \__tilings_use_tile:nn {#1}{#2}
930 }
```

**\tiling\_path** 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.

```
931 \tikzset{
932   save~ tiling~ path/.code={
933     \tikz@addmode{
```

Get the current path.

```
934     \pgfsyssoftpath@getcurrentpath\l__tilings_tmpa_tl
```

Normalise the path and save.

```
935     \__tilings_normalise_path:N \l__tilings_tmpa_tl
936     \tl_gclear_new:c {g__tilings_side_#1_tl}
937     \tl_gset_eq:cN {g__tilings_side_#1_tl} \l__tilings_tmpa_tl
```

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

```
938     \tl_set:Nx \l__tilings_tmpb_tl {\str_uppercase:n {#1}}
```

Reverse the path, and relocate to the interval  $[0, 1]$ .

```
939     \spath_reverse:N \l__tilings_tmpa_tl
940     \spath_transform:Nnnnnn \l__tilings_tmpa_tl {-1} {0} {0} {-1} {1} {0}
941     \tl_gclear_new:c {g__tilings_side_ \tl_use:N \l__tilings_tmpb_tl _tl}
```

```

942     \tl_gset_eq:cN {g__tilings_side_ \tl_use:N \l__tilings_tmpb_tl _tl} \l__tilings_tmpa_tl
943   }
944 },
945 clone~ tiling~ side~ path/.style~ 2~ args={
946   spath/set~ name=tiling~ side,
947   spath/clone~ global={#1}{#2}
948 },
949 flip~ tile/.code={
950   \tl_set:Nn \l__tilings_tmpa_tl {#1}
951   \tl_set:Nn \l__tilings_tmpb_tl {true}
952   \bool_set:Nn \l__tilings_cw_bool {\tl_if_eq_p:NN \l__tilings_tmpa_tl \l__tilings_tmpb_tl}
953 },
954 flip~ tile/.default={true},
955 spath/prefix/tiling~side/.style={
956   spath/set~ prefix=g__tilings_side_,
957 },
958 spath/suffix/tiling~side/.style={
959   spath/set~ suffix=_tl,
960 },
961 clone~ tile~ path/.style~ 2~ args={
962   spath/set~ name=tiling~tile,
963   spath/clone~ global={#1}{#2}
964 },
965 spath/prefix/tiling~tile/.style={
966   spath/set~ prefix=g__tilings_tile_,
967 },
968 spath/suffix/tiling~tile/.style={
969   spath/set~ suffix=_tl,
970 },
971 expand~ key/.code={
972   \exp_args:NV \pgfkeysalso #1
973 }
974 }

```

(End definition for save tiling path. This function is documented on page ??.)

## 2.5 Lindenmayer System

This is an implementation of the Lindenmayer System description of Penrose and other 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<sup>A</sup>T<sub>E</sub>X3 it is reimplemented in that.

`\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.

```

975 \cs_new_nopar:Npn \__tilings_make_lms:Nnnn #1#2#3#4
976 {
977   \group_begin:

```

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

```

978   \tl_set:Nn \l__tilings_tmpb_tl {#4}

```

We repeat the specified number of times.

```

979   \prg_replicate:nn {#3} {

```

Duplicate the current state.

```

980     \tl_set_eq:NN \l__tilings_tmpa_tl \l__tilings_tmpb_tl

```

Clear the receiving token list.

```

981     \tl_clear:N \l__tilings_tmpb_tl

```

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

```
982 \tl_map_inline:Nn \l__tilings_tmpa_tl
983 {
```

If a rule exists, copy that.

```
984 \tl_set:Nx \l__tilings_action_lms_tl {\tl_head:n {##1}}
985 \tl_set:Nx \l__tilings_parameters_lms_tl {\tl_tail:n {##1}}
986 \prop_if_in:cVTF {g__tilings_#2_lms_rule_prop} \l__tilings_action_lms_tl
987 {
988 \prop_get:cVN {g__tilings_#2_lms_rule_prop} \l__tilings_action_lms_tl \l__tilings_tmpc_tl
989 \tl_put_right:Nx \l__tilings_tmpb_tl {\tl_use:N \l__tilings_tmpc_tl}
990 % {\prop_item:cn {g__tilings_#2_lms_rule_prop} {##1} }
991 }
992 {
```

Otherwise, just copy the token.

```
993 \tl_if_single:nTF {##1}
994 {
995 \tl_put_right:Nn \l__tilings_tmpb_tl {##1}
996 }
997 {
998 \tl_put_right:Nn \l__tilings_tmpb_tl {{##1}}
999 }
1000 }
1001 }
1002 }
```

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

```
1003 \tl_set:Nn \l__tilings_tmpa_tl {
1004 \group_end:
1005 \tl_set:Nn #1
1006 }
1007 \tl_put_right:Nx \l__tilings_tmpa_tl {{\tl_use:N \l__tilings_tmpb_tl}}
1008 \tl_use:N \l__tilings_tmpa_tl
1009 }
1010 \cs_generate_variant:Nn \__tilings_make_lms:Nnnn {Nnnx}
```

*(End definition for \\_\_tilings\_make\_lms:Nnnn.)*

`voke_lms:nn` This carries out the actions specified by the resulting rules.

```
1011 \cs_new_nopar:Npn \__tilings_invoke_lms:nn #1#2
1012 {
1013 \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.

```
1014 \tl_map_inline:nn {#1} {
1015 \tl_set:Nx \l__tilings_action_lms_tl {\tl_head:n {##1}}
1016 \tl_set:Nx \l__tilings_parameters_lms_tl {\tl_tail:n {##1}}
1017 \prop_if_in:cVTF {g__tilings_#2_lms_action_prop} \l__tilings_action_lms_tl
1018 {
1019 \prop_item:cV {g__tilings_#2_lms_action_prop} \l__tilings_action_lms_tl
1020 }
1021 {
1022 \prop_if_in:cVT {g__tilings_default_lms_action_prop} \l__tilings_action_lms_tl
1023 {
1024 \prop_item:cV {g__tilings_default_lms_action_prop} \l__tilings_action_lms_tl
1025 }
1026 }
1027 }
1028 \group_end:
1029 }
1030 \cs_generate_variant:Nn \__tilings_invoke_lms:nn {Vn}
```

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

We need some parameters.

```
1031 \dim_new:N \l__tilings_step_dim
1032 \dim_set:Nn \l__tilings_step_dim {1cm}
```

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

```
1033 \prop_new:N \g__tilings_default_lms_action_prop
1034 \prop_gput:Nnn \g__tilings_default_lms_action_prop {} {\group_begin:}
1035 \prop_gput:Nnn \g__tilings_default_lms_action_prop {} {\group_end:}
1036 \prop_gput:Nnn \g__tilings_default_lms_action_prop {f}
1037 {\pgftransformxshift{\l__tilings_step_dim}}
1038 \prop_gput:Nnn \g__tilings_default_lms_action_prop {b}
1039 {\pgftransformxshift{-\l__tilings_step_dim}}
```

Holds a list of the tiles that actually draw for each tile set

```
1040 \prop_new:N \g__tilings_drawables_lms_prop
```

We keep track of the number of tiles.

```
1041 \int_new:N \g__tilings_tile_int
1042 \int_new:N \g__tilings_tiles_int
```

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

```
1043 \cs_new_protected_nopar:Npn \__tilings_tiling_decomposition:nnnn #1#2#3#4
1044 {
1045   \group_begin:
1046   \tikzset{
1047     every~ #2~ decomposition/.try,
1048     #1
1049   }
1050   \__tilings_make_lms:Nnnx \l__tilings_tmpa_tl {#2} {#3} {#4}
1051   \__tilings_count_lms:Vn \l__tilings_tmpa_tl {#2}
1052   \int_gzero:N \g__tilings_tile_int
1053   \__tilings_invoke_lms:Vn \l__tilings_tmpa_tl {#2}
1054   \group_end:
1055 }
1056 \cs_new_protected_nopar:Npn \__tilings_tiling_decomposition:nnn #1#2#3
1057 {
1058   \__tilings_tiling_decomposition:nnnn {}{#1}{#2}{#3}
1059 }
1060 \cs_generate_variant:Nn \__tilings_tiling_decomposition:nnn {VVV}
1061
1062 \NewDocumentCommand \TilingDecomposition { O{} m m m }
1063 {
1064   \__tilings_tiling_decomposition:nnnn {#1}{#2}{#3}{#4}
1065 }
1066
1067 \tikzset{
1068   pics/decomposition/.style~ n~ args={3}{
1069     code={
1070       \__tilings_tiling_decomposition:nnn {#1}{#2}{#3}
1071     }
1072   }
1073 }
```

**count\_lms:nn** This counts the number of tiles in the string.

```
1074 \cs_new_nopar:Npn \__tilings_count_lms:nn #1#2
1075 {
1076   \group_begin:
1077   \int_gzero:N \g__tilings_tiles_int
```

```

1078 \prop_get:NnNT \g__tilings_drawables_lms_prop {#2} \l__tilings_tmpa_tl
1079 {
1080   \tl_map_variable:nNn {#1} \l__tilings_tmpb_tl
1081   {
1082     \tl_set:Nx \l__tilings_tmpb_tl {\tl_head:N \l__tilings_tmpb_tl}
1083     \bool_do_while:nn
1084     {
1085       !\tl_if_empty_p:N \l__tilings_tmpb_tl
1086       &&
1087       \tl_if_head_is_group_p:V \l__tilings_tmpb_tl
1088     }
1089     {
1090       \tl_set:Nx \l__tilings_tmpb_tl {\tl_head:N \l__tilings_tmpb_tl}
1091     }
1092     \tl_if_in:NVT \l__tilings_tmpa_tl \l__tilings_tmpb_tl
1093     {
1094       \int_gincr:N \g__tilings_tiles_int
1095     }
1096   }
1097 }
1098 \group_end:
1099 }
1100 \cs_generate_variant:Nn \__tilings_count_lms:nn {Vn}

```

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

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

```

1101 \tikzset{
1102   tiling~ step/.code={
1103     \dim_set:Nn \l__tilings_step_dim {#1}
1104   }
1105 }
1106 \ExplSyntaxOff
1107 \RequirePackage{tikz}
1108 \usetikzlibrary{tilings}
1109 \ProvidesFile {tikzlibrarytilings.penrose.code.tex}
1110 [2023/06/01 v2.0 TikZ pics for Penrose tiles]

```

Create the pre-defined tile shapes.

- Thin Rhombus.

```

1111 \DefineTile{thin rhombus}{a A B b}
1112 {
1113   {0 , 0}
1114   {cosd(18) , sind(18)}
1115   {2*cosd(18) , 0}
1116   {cosd(18) , -sind(18)}
1117 }

```

- Thick Rhombus.

```

1118 \DefineTile{thick rhombus}{B a A b}
1119 {
1120   {0 , 0}
1121   {cosd(36) , sind(36)}
1122   {2*cosd(36) , 0}
1123   {cosd(36) , -sind(36)}
1124 }

```

- Dart.

```

1125 \DefineTile{dart}{c a A C}
1126 {
1127   {0 , 0}
1128   {2*sind(18)*cosd(108) , 2*sind(18)*sind(108)}
1129   {2*sind(18) , 0}
1130   {2*sind(18)*cosd(108) , -2*sind(18)*sind(108)}
1131 }

```

- Kite.

```

1132 \DefineTile{kite}{a c C A}
1133 {
1134   {0 , 0}
1135   {cosd(36) , sind(36)}
1136   {1 , 0}
1137   {cosd(36) , -sind(36)}
1138 }

```

- Golden Triangle.

```

1139 \DefineTile{golden triangle}{a c b}
1140 {
1141   {0 , 0}
1142   {cosd(18) , sind(18)}
1143   {cosd(18) , -sind(18)}
1144 }

```

- Reverse Golden Triangle.

```

1145 \DefineTile {reverse golden triangle}{B C A}
1146 {
1147   {0 , 0}
1148   {cosd(18) , sind(18)}
1149   {cosd(18) , -sind(18)}
1150 }

```

- Golden Gnomon

```

1151 \DefineTile {golden gnomon}{C b A}
1152 {
1153   {0 , 0}
1154   {cosd(36) , sind(36)}
1155   {2*cosd(36) , 0}
1156 }

```

- Reverse Golden Gnomon

```

1157 \DefineTile {reverse golden gnomon}{a B c}
1158 {
1159   {0 , 0}
1160   {2*cosd(36) , 0}
1161   {cosd(36) , -sind(36)}
1162 }

```

- Primary Pentagon (pentagon 5)

```

1163 \DefineTile {pentagon 5}{a a a a a}
1164 {

```

```

1165     {0 , 0}
1166     {cosd(108) , sind(108)}
1167     {1+cosd(72)+cosd(144) , sind(72)+sind(144)}
1168     {1+cosd(72) , sind(72)}
1169     {1 , 0}
1170     }

```

- Secondary Pentagon (pentagon 3)

```

1171     \DefineTile {pentagon 3}{A b a a b}
1172     {
1173     {0 , 0}
1174     {cosd(108) , sind(108)}
1175     {1+cosd(72)+cosd(144) , sind(72)+sind(144)}
1176     {1+cosd(72) , sind(72)}
1177     {1 , 0}
1178     }

```

- Tertiary Pentagon (pentagon 2)

```

1179     \DefineTile {pentagon 2}{d A e c A}
1180     {
1181     {0 , 0}
1182     {cosd(108) , sind(108)}
1183     {1+cosd(72)+cosd(144) , sind(72)+sind(144)}
1184     {1+cosd(72) , sind(72)}
1185     {1 , 0}
1186     }

```

- Pentagon

```

1187     \DefineTile {pentagram}{C E C E C E C E}
1188     {
1189     {1 , 0}
1190     {1-cosd(36) , -sind(36)}
1191     {1-cosd(36)-cosd(108) , -sind(36)-sind(108)}
1192     {cosd(108) , -sind(108)}
1193     {-1+3*cosd(108)+cosd(36) , -sind(36)-sind(108)}
1194     {-1+2*cosd(108)+cosd(36) , -sind(36)}
1195     {-1+2*cosd(108) , 0}
1196     {2*cosd(108) , 0}
1197     {cosd(108) , sind(108)}
1198     {0 , 0}
1199     }

```

- Boat

```

1200     \DefineTile {boat}{C E C E B D B}
1201     {
1202     {-1+2*cosd(108) , 0}
1203     {2*cosd(108) , 0}
1204     {cosd(108) , sind(108)}
1205     {0 , 0}
1206     {1 , 0}
1207     {1-cosd(36) , -sind(36)}
1208     {-1+2*cosd(108)+cosd(36) , -sind(36)}
1209     }

```

- Diamond.

```

1210 \DefineTile {diamond}{D B B D}
1211 {
1212   {0 , 0}
1213   {cosd(18) , sind(18)}
1214   {2*cosd(18) , 0}
1215   {cosd(18) , -sind(18)}
1216 }

```

Place the arcs

```

1217 \tikzset{
1218   every thin rhombus before path/.code={
1219     \path[every circle arc/.try] (-edge a end) circle[radius=1/4];
1220     \path[every long arc/.try] (-edge b start) circle[radius=1/4];
1221   },
1222   every thick rhombus before path/.code={
1223     \path[every circle arc/.try] (-edge a end) circle[radius=1/4];
1224     \path[every long arc/.try] (-edge B start) circle[radius=3/4];
1225   },
1226   every kite before path/.code={
1227     \path[every circle arc/.try] (-edge a start) circle[radius=2/(sqrt(5)+1)];
1228     \path[every long arc/.try] (-edge c end) circle[radius=2/(3+sqrt(5))];
1229   },
1230   every dart before path/.code={
1231     \path[every circle arc/.try] (-edge a end) circle[radius=1 - 2/(sqrt(5)+1)];
1232     \path[every long arc/.try] (-edge c start) circle[radius=2/(sqrt(5)+1) - 2/(3+sqrt(5))];
1233   }
1234 }

```

Now bake the tiles.

```

1235 \BakeTile {thin rhombus}
1236 \BakeTile {thick rhombus}
1237 \BakeTile {dart}
1238 \BakeTile {kite}
1239 \BakeTile {golden triangle}
1240 \BakeTile {reverse golden triangle}
1241 \BakeTile {golden gnomon}
1242 \BakeTile {reverse golden gnomon}
1243 \BakeTile {pentagon 5}
1244 \BakeTile {pentagon 3}
1245 \BakeTile {pentagon 2}
1246 \BakeTile {pentagram}
1247 \BakeTile {boat}
1248 \BakeTile {diamond}

```

## 2.6 Lindenmayer System

```
1249 \ExplSyntaxOn
```

These are the rules for generating rhombus tilings with the Lindenmayer System procedure.

```

1250 \prop_new:N \g__tilings_rhombus_lms_rule_prop
1251 \prop_gput:Nnn \g__tilings_rhombus_lms_rule_prop {T} {[f*sT] [f>g]}
1252 \prop_gput:Nnn \g__tilings_rhombus_lms_rule_prop {t} {[f_st] [f>G]}
1253 \prop_gput:Nnn \g__tilings_rhombus_lms_rule_prop {G} {[f+sG] [sf>g] [sf*sT]}
1254 \prop_gput:Nnn \g__tilings_rhombus_lms_rule_prop {g} {[f-sg] [sf>G] [sf_st]}

```

These are the rules for generating kite and dart tilings.

```

1255 \prop_new:N \g__tilings_kite_lms_rule_prop
1256 \prop_gput:Nnn \g__tilings_kite_lms_rule_prop {T} {[f*sT] [f>st] [+sg]}
1257 \prop_gput:Nnn \g__tilings_kite_lms_rule_prop {t} {[f_st] [f>sT] [-sG]}
1258 \prop_gput:Nnn \g__tilings_kite_lms_rule_prop {G} {[f**sG] [sT]}
1259 \prop_gput:Nnn \g__tilings_kite_lms_rule_prop {g} {[f-_sg] [st]}

```

These are the rules for generating pentagon tilings.



```

1260 \prop_new:N \g__tilings_pentagon_lms_rule_prop
1261 \prop_gput:Nnn \g__tilings_pentagon_lms_rule_prop {P}
1262 {[s>P][1sF+Q][1+sF+Q][1*sF+Q][1-sF+Q][1_sF+Q]} % pentagon 5
1263 \prop_gput:Nnn \g__tilings_pentagon_lms_rule_prop {Q}
1264 {[s>P][1+sFR][1*sFR][1-sF+Q][1_sF+Q][1sF+Q][>fsD]} % pentagon 3
1265 \prop_gput:Nnn \g__tilings_pentagon_lms_rule_prop {R}
1266 {[s>P][1-sF+Q][1+sFR][1*sFR][1_sFR][1sFR][>fsD][>fsD]} % pentagon 2
1267 \prop_gput:Nnn \g__tilings_pentagon_lms_rule_prop {G}
1268 {
1269   [s>G]
1270   [se[>d+R][e1B]]
1271   [+se[>d+R][e1B]]
1272   [-se[>d+R][e1B]]
1273   [*se[>d+R][e1B]]
1274   [_se[>d+R][e1B]]
1275 } % pentagram
1276 \prop_gput:Nnn \g__tilings_pentagon_lms_rule_prop {B}
1277 {
1278   [s>G]
1279   [se[>d+R][e1B]]
1280   [+se[>d+R][e1B]]
1281   [-se[>d+R][e1B]]
1282 } % boat
1283 \prop_gput:Nnn \g__tilings_pentagon_lms_rule_prop {D}
1284 {[s>d+R][s>eG][se1B]} % diamond

```

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

```

1285 \prop_gset_eq:NN \g__tilings_rtriangle_lms_rule_prop
1286 \g__tilings_rhombus_lms_rule_prop
1287 \prop_gset_eq:NN \g__tilings_ktriangle_lms_rule_prop
1288 \g__tilings_kite_lms_rule_prop

```

These are the lists of tokens that actually draw things

```

1289 \prop_gput:Nnn \g__tilings_drawables_lms_prop {rhombus} {TG}
1290 \prop_gput:Nnn \g__tilings_drawables_lms_prop {kite} {Tg}
1291 \prop_gput:Nnn \g__tilings_drawables_lms_prop {rtriangle} {TtGg}
1292 \prop_gput:Nnn \g__tilings_drawables_lms_prop {ktriangle} {TtGg}
1293 \prop_gput:Nnn \g__tilings_drawables_lms_prop {pentagon} {PQRGBD}

```

These hold the various actions.

```

1294 \prop_new:N \g__tilings_rhombus_lms_action_prop
1295 \prop_new:N \g__tilings_kite_lms_action_prop
1296 \prop_new:N \g__tilings_rtriangle_lms_action_prop
1297 \prop_new:N \g__tilings_ktriangle_lms_action_prop
1298 \prop_new:N \g__tilings_pentagon_lms_action_prop

```

The rhombus rules need a variety of turns.

```

1299 \prop_gput:Nnn \g__tilings_rhombus_lms_action_prop {+}
1300 {\pgftransformrotate{144}}
1301 \prop_gput:Nnn \g__tilings_rhombus_lms_action_prop {*}
1302 {\pgftransformrotate{108}}
1303 \prop_gput:Nnn \g__tilings_rhombus_lms_action_prop {-}
1304 {\pgftransformrotate{216}}
1305 \prop_gput:Nnn \g__tilings_rhombus_lms_action_prop {_}
1306 {\pgftransformrotate{252}}
1307 \prop_gput:Nnn \g__tilings_rhombus_lms_action_prop {>}
1308 {\pgftransformrotate{180}}
1309 \prop_gput:Nnn \g__tilings_rhombus_lms_action_prop {s} {
1310   \fp_set:Nn \l__tilings_tmpa_fp { 2 * sind(18) * \l__tilings_step_dim }
1311   \dim_set:Nn \l__tilings_step_dim {\fp_to_dim:N \l__tilings_tmpa_fp}
1312 }

```

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

```

1313 \prop_gset_eq:NN \g__tilings_rtriangle_lms_action_prop

```

1314 \g\_\_tilings\_rhombus\_lms\_action\_prop

Now we do the actions that actually draw something.

1315 \prop\_gput:Nnn \g\_\_tilings\_rhombus\_lms\_action\_prop {T} {  
1316 \group\_begin:

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

1317 \int\_gincr:N \g\_\_tilings\_tile\_int

Set up the position, size, and angle correctly.

1318 \pgftransformrotate{198}  
1319 \fp\_set:Nn \l\_\_tilings\_tmpa\_fp {\l\_\_tilings\_step\_dim\*2\*cosd(18)}  
1320 \pgftransformxshift{-\fp\_to\_dim:N \l\_\_tilings\_tmpa\_fp}  
1321 \fp\_set:Nn \l\_\_tilings\_tmpa\_fp {\l\_\_tilings\_step\_dim/(1cm)}  
1322 \pgftransformscale{\fp\_use:N \l\_\_tilings\_tmpa\_fp}

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

1323 \tl\_set:Nx \l\_\_tilings\_tmpc\_tl  
1324 {  
1325 {\int\_use:N \g\_\_tilings\_tile\_int}  
1326 {\int\_use:N \g\_\_tilings\_tiles\_int}  
1327 }  
1328 \UseTile[  
1329 every~ tile/.try,  
1330 every~ thin~ rhombus/.try,  
1331 tile~ \int\_use:N \g\_\_tilings\_tile\_int/.try,  
1332 tile/.try/.expand~ once=\l\_\_tilings\_tmpc\_tl  
1333 ]{thin~rhombus}  
1334 \group\_end:  
1335 }

Same for the thick rhombus.

1336 \prop\_gput:Nnn \g\_\_tilings\_rhombus\_lms\_action\_prop {G} {  
1337 \group\_begin:  
1338 \int\_gincr:N \g\_\_tilings\_tile\_int  
1339 \fp\_set:Nn \l\_\_tilings\_tmpa\_fp {\l\_\_tilings\_step\_dim/(1cm)/(2\*cosd(36))}  
1340 \pgftransformscale{\fp\_use:N \l\_\_tilings\_tmpa\_fp}  
1341 \tl\_set:Nx \l\_\_tilings\_tmpc\_tl  
1342 {  
1343 {\int\_use:N \g\_\_tilings\_tile\_int}  
1344 {\int\_use:N \g\_\_tilings\_tiles\_int}  
1345 }  
1346 \UseTile[  
1347 every~ tile/.try,  
1348 every~ thick~ rhombus/.try,  
1349 tile~ \int\_use:N \g\_\_tilings\_tile\_int/.try,  
1350 tile/.try/.expand~ once=\l\_\_tilings\_tmpc\_tl  
1351 ]{thick~rhombus}  
1352 \group\_end:  
1353 }

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

1354 \prop\_gput:Nnn \g\_\_tilings\_kite\_lms\_action\_prop {+} {\pgftransformrotate{36}}  
1355 \prop\_gput:Nnn \g\_\_tilings\_kite\_lms\_action\_prop {\*} {\pgftransformrotate{108}}  
1356 \prop\_gput:Nnn \g\_\_tilings\_kite\_lms\_action\_prop {-} {\pgftransformrotate{-36}}  
1357 \prop\_gput:Nnn \g\_\_tilings\_kite\_lms\_action\_prop {\_}  
1358 {\pgftransformrotate{-108}}  
1359 \prop\_gput:Nnn \g\_\_tilings\_kite\_lms\_action\_prop {>} {\pgftransformrotate{180}}  
1360 \prop\_gput:Nnn \g\_\_tilings\_kite\_lms\_action\_prop {s} {  
1361 \fp\_set:Nn \l\_\_tilings\_tmpa\_fp { 2 \* sind(18) \* \l\_\_tilings\_step\_dim }  
1362 \dim\_set:Nn \l\_\_tilings\_step\_dim {\fp\_to\_dim:N \l\_\_tilings\_tmpa\_fp}  
1363 }  
1364 \prop\_gset\_eq:NN \g\_\_tilings\_ktriangle\_lms\_action\_prop  
1365 \g\_\_tilings\_kite\_lms\_action\_prop

```

1366 \prop_gput:Nnn \g__tilings_kite_lms_action_prop {T} {
1367   \group_begin:
1368   \int_gincr:N \g__tilings_tile_int
1369   \pgftransformrotate{36}
1370   \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim/(1cm)}
1371   \pgftransformscale{\fp_use:N \l__tilings_tmpa_fp}
1372   \tl_set:Nx \l__tilings_tmpc_tl
1373   {
1374     {\int_use:N \g__tilings_tile_int}
1375     {\int_use:N \g__tilings_tiles_int}
1376   }
1377   \UseTile[
1378     every~ tile/.try,
1379     every~ kite/.try,
1380     tile~ \int_use:N \g__tilings_tile_int/.try,
1381     tile/.try/.expand~ once=\l__tilings_tmpc_tl
1382   ]{kite}
1383   \group_end:
1384 }

```

```

1385 \prop_gput:Nnn \g__tilings_kite_lms_action_prop {g} {
1386   \group_begin:
1387   \int_gincr:N \g__tilings_tile_int
1388   \pgftransformrotate{144}
1389   \pgftransformxshift{-\l__tilings_step_dim * 2 * sin(18)}
1390   \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim/(1cm)}
1391   \pgftransformscale{\fp_use:N \l__tilings_tmpa_fp}
1392   \tl_set:Nx \l__tilings_tmpc_tl
1393   {
1394     {\int_use:N \g__tilings_tile_int}
1395     {\int_use:N \g__tilings_tiles_int}
1396   }
1397   \UseTile[
1398     every~ tile/.try,
1399     every~ dart/.try,
1400     tile~ \int_use:N \g__tilings_tile_int/.try,
1401     tile/.try/.expand~ once=\l__tilings_tmpc_tl
1402   ]{dart}
1403   \group_end:
1404 }

```

Now we set up the actions for the triangle variations.

```

1405 \prop_gput:Nnn \g__tilings_rtriangle_lms_action_prop {T} {
1406   \group_begin:
1407   \int_gincr:N \g__tilings_tile_int
1408   \pgftransformrotate{18}
1409   \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim/(1cm)}
1410   \pgftransformscale{\fp_use:N \l__tilings_tmpa_fp}
1411   \tl_set:Nx \l__tilings_tmpc_tl
1412   {
1413     {\int_use:N \g__tilings_tile_int}
1414     {\int_use:N \g__tilings_tiles_int}
1415   }
1416   \UseTile[
1417     every~ tile/.try,
1418     every~ reverse~ golden~ triangle/.try,
1419     tile~ \int_use:N \g__tilings_tile_int/.try,
1420     tile/.try/.expand~ once=\l__tilings_tmpc_tl
1421   ]{reverse~ golden~ triangle}
1422   \group_end:
1423 }
1424 \prop_gput:Nnn \g__tilings_rtriangle_lms_action_prop {t} {

```

```

1425 \group_begin:
1426 \int_gincr:N \g__tilings_tile_int
1427 \pgftransformrotate{-18}
1428 \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim/(1cm)}
1429 \pgftransformscale{\fp_use:N \l__tilings_tmpa_fp}
1430 \tl_set:Nx \l__tilings_tmpc_tl
1431 {
1432   {\int_use:N \g__tilings_tile_int}
1433   {\int_use:N \g__tilings_tiles_int}
1434 }
1435 \tl_set:Nx \l__tilings_tmpc_tl
1436 {
1437   {\int_use:N \g__tilings_tile_int}
1438   {\int_use:N \g__tilings_tiles_int}
1439 }
1440 \UseTile[
1441   every~ tile/.try,
1442   every~ golden~ triangle/.try,
1443   tile~ \int_use:N \g__tilings_tile_int/.try,
1444   tile/.try/.expand~ once=\l__tilings_tmpc_tl
1445 ]{golden~ triangle}
1446 \group_end:
1447 }

1448 \prop_gput:Nnn \g__tilings_rtriangle_lms_action_prop {G} {
1449   \group_begin:
1450   \int_gincr:N \g__tilings_tile_int
1451   \pgftransformrotate{180}
1452   \pgftransformxshift{-\l__tilings_step_dim}
1453   \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim/(1cm)/(2*cosd(36))}
1454   \pgftransformscale{\fp_use:N \l__tilings_tmpa_fp}
1455   \tl_set:Nx \l__tilings_tmpc_tl
1456   {
1457     {\int_use:N \g__tilings_tile_int}
1458     {\int_use:N \g__tilings_tiles_int}
1459   }
1460   \UseTile[
1461     every~ tile/.try,
1462     every~ reverse~ golden~ gnomon/.try,
1463     tile~ \int_use:N \g__tilings_tile_int/.try,
1464     tile/.try/.expand~ once=\l__tilings_tmpc_tl
1465   ]{reverse~ golden~ gnomon}
1466   \group_end:
1467 }

1468 \prop_gput:Nnn \g__tilings_rtriangle_lms_action_prop {g} {
1469   \group_begin:
1470   \int_gincr:N \g__tilings_tile_int
1471   \pgftransformrotate{180}
1472   \pgftransformxshift{-\l__tilings_step_dim}
1473   \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim/(1cm)/(2*cosd(36))}
1474   \pgftransformscale{\fp_use:N \l__tilings_tmpa_fp}
1475   \tl_set:Nx \l__tilings_tmpc_tl
1476   {
1477     {\int_use:N \g__tilings_tile_int}
1478     {\int_use:N \g__tilings_tiles_int}
1479   }
1480   \UseTile[
1481     every~ tile/.try,
1482     every~ golden~ gnomon/.try,
1483     tile~ \int_use:N \g__tilings_tile_int/.try,
1484     tile/.try/.expand~ once=\l__tilings_tmpc_tl
1485   ]{golden~ gnomon}

```

```

1486 \group_end:
1487 }

1488 \prop_gput:Nnn \g__tilings_ktriangle_lms_action_prop {T} {
1489 \group_begin:
1490 \int_gincr:N \g__tilings_tile_int
1491 \pgftransformrotate{18}
1492 \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim/(1cm)}
1493 \pgftransformscale{\fp_use:N \l__tilings_tmpa_fp}
1494 \tl_set:Nx \l__tilings_tmpc_tl
1495 {
1496 {\int_use:N \g__tilings_tile_int}
1497 {\int_use:N \g__tilings_tiles_int}
1498 }
1499 \UseTile[
1500 every~ tile/.try,
1501 every~ reverse~ golden~ triangle/.try,
1502 tile~ \int_use:N \g__tilings_tile_int/.try,
1503 tile/.try/.expand~ once=\l__tilings_tmpc_tl
1504 ]{reverse~ golden~ triangle}
1505 \group_end:
1506 }

1507 \prop_gput:Nnn \g__tilings_ktriangle_lms_action_prop {t} {
1508 \group_begin:
1509 \int_gincr:N \g__tilings_tile_int
1510 \pgftransformrotate{-18}
1511 \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim/(1cm)}
1512 \pgftransformscale{\fp_use:N \l__tilings_tmpa_fp}
1513 \tl_set:Nx \l__tilings_tmpc_tl
1514 {
1515 {\int_use:N \g__tilings_tile_int}
1516 {\int_use:N \g__tilings_tiles_int}
1517 }
1518 \UseTile[
1519 every~ tile/.try,
1520 every~ golden~ triangle/.try,
1521 tile~ \int_use:N \g__tilings_tile_int/.try,
1522 tile/.try/.expand~ once=\l__tilings_tmpc_tl
1523 ]{golden~ triangle}
1524 \group_end:
1525 }

1526 \prop_gput:Nnn \g__tilings_ktriangle_lms_action_prop {G} {
1527 \group_begin:
1528 \int_gincr:N \g__tilings_tile_int
1529 \pgftransformrotate{180}
1530 \pgftransformxshift{-\l__tilings_step_dim}
1531 \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim/(1cm)/(2*cosd(36))}
1532 \pgftransformscale{\fp_use:N \l__tilings_tmpa_fp}
1533 \tl_set:Nx \l__tilings_tmpc_tl
1534 {
1535 {\int_use:N \g__tilings_tile_int}
1536 {\int_use:N \g__tilings_tiles_int}
1537 }
1538 \UseTile[
1539 every~ tile/.try,
1540 every~ reverse~ golden~ gnomon/.try,
1541 tile~ \int_use:N \g__tilings_tile_int/.try,
1542 tile/.try/.expand~ once=\l__tilings_tmpc_tl
1543 ]{reverse~ golden~ gnomon}
1544 \group_end:
1545 }

```

```

1546 \prop_gput:Nnn \g__tilings_ktriangle_lms_action_prop {g} {
1547   \group_begin:
1548   \int_gincr:N \g__tilings_tile_int
1549   \pgftransformrotate{180}
1550   \pgftransformxshift{-\l__tilings_step_dim}
1551   \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim/(1cm)/(2*cosd(36))}
1552   \pgftransformscale{\fp_use:N \l__tilings_tmpa_fp}
1553   \tl_set:Nx \l__tilings_tmpe_tl
1554   {
1555     {\int_use:N \g__tilings_tile_int}
1556     {\int_use:N \g__tilings_tiles_int}
1557   }
1558   \UseTile[
1559     every~ tile/.try,
1560     every~ golden~ gnomon/.try,
1561     tile~ \int_use:N \g__tilings_tile_int/.try,
1562     tile/.try/.expand~ once=\l__tilings_tmpe_tl
1563   ]{golden~ gnomon}
1564   \group_end:
1565 }

```

Now we do the same for the pentagonal tilings.

The rules need a variety of turns.

```

1566 \int_new:N \l__tilings_pentagon_parity_int
1567 \seq_new:N \l__tilings_pentagon_parity_seq
1568 \seq_set_from_clist:Nn \l__tilings_pentagon_parity_seq {odd,even}
1569 \prop_gput:Nnn \g__tilings_pentagon_lms_action_prop {1} {
1570   \int_set:Nn \l__tilings_pentagon_parity_int
1571   {3 - \l__tilings_pentagon_parity_int}
1572 }
1573 \tikzset{
1574   every~ pentagon~ decomposition/.code={%
1575     \int_set:Nn \l__tilings_pentagon_parity_int {2}
1576   }
1577 }
1578 \prop_gput:Nnn \g__tilings_pentagon_lms_action_prop {+}
1579 {\pgftransformrotate{72}}
1580 \prop_gput:Nnn \g__tilings_pentagon_lms_action_prop {*}
1581 {\pgftransformrotate{144}}
1582 \prop_gput:Nnn \g__tilings_pentagon_lms_action_prop {-}
1583 {\pgftransformrotate{288}}
1584 \prop_gput:Nnn \g__tilings_pentagon_lms_action_prop {_}
1585 {\pgftransformrotate{216}}
1586 \prop_gput:Nnn \g__tilings_pentagon_lms_action_prop {>}
1587 {\pgftransformrotate{180}}
1588 \prop_gput:Nnn \g__tilings_pentagon_lms_action_prop {|}
1589 {\pgftransformxscale{-1}}

```

The scale factor is different.

```

1590 \prop_gput:Nnn \g__tilings_pentagon_lms_action_prop {s} {
1591   \fp_set:Nn \l__tilings_tmpa_fp
1592   {
1593     1/(2 + 2 * cosd(72) ) * \l__tilings_step_dim
1594   }
1595   \dim_set:Nn \l__tilings_step_dim {\fp_to_dim:N \l__tilings_tmpa_fp}
1596 }

```

And we tend to work better vertically.

```

1597 \prop_gput:Nnn \g__tilings_pentagon_lms_action_prop {f} {
1598   \fp_set:Nn \l__tilings_tmpa_fp { tand(54)/2 * \l__tilings_step_dim }
1599   \pgftransformyshift{\fp_to_dim:N \l__tilings_tmpa_fp}
1600 }
1601 \prop_gput:Nnn \g__tilings_pentagon_lms_action_prop {F} {

```

```

1602 \fp_set:Nn \l__tilings_tmpa_fp { tand(54) * \l__tilings_step_dim }
1603 \pgftransformyshift{\fp_to_dim:N \l__tilings_tmpa_fp}
1604 }
1605 \prop_gput:Nnn \g__tilings_pentagon_lms_action_prop {d} {
1606 \fp_set:Nn \l__tilings_tmpa_fp
1607 {
1608 (tand(54)/2 - tand(72)/2 + sind(36) ) * \l__tilings_step_dim
1609 }
1610 \pgftransformyshift{\fp_to_dim:N \l__tilings_tmpa_fp}
1611 }
1612 \prop_gput:Nnn \g__tilings_pentagon_lms_action_prop {e} {
1613 \fp_set:Nn \l__tilings_tmpa_fp
1614 {
1615 tand(54) * cosd(36) * \l__tilings_step_dim
1616 }
1617 \pgftransformyshift{\fp_to_dim:N \l__tilings_tmpa_fp}
1618 }
1619 \prop_gput:Nnn \g__tilings_pentagon_lms_action_prop {P} {
1620 \group_begin:
1621 \int_gincr:N \g__tilings_tile_int
1622 \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim/2}
1623 \pgftransformxshift{-\fp_to_dim:N \l__tilings_tmpa_fp}
1624 \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim*tand(54)/2}
1625 \pgftransformyshift{-\fp_to_dim:N \l__tilings_tmpa_fp}
1626 \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim/(1cm)}
1627 \pgftransformscale{\fp_use:N \l__tilings_tmpa_fp}
1628 \tl_set:Nx \l__tilings_tmpe_tl
1629 {
1630 {\int_use:N \g__tilings_tile_int}
1631 {\int_use:N \g__tilings_tiles_int}
1632 }
1633 \UseTile[
1634 every~ tile/.try,
1635 every~ pentagon/.try,
1636 every~
1637 \seq_item:Nn \l__tilings_pentagon_parity_seq
1638 {\l__tilings_pentagon_parity_int}
1639 \space pentagon/.try,
1640 every~ pentagon~ 5/.try,
1641 tile~ \int_use:N \g__tilings_tile_int/.try,
1642 tile/.try/.expand~ once=\l__tilings_tmpe_tl
1643 ]{pentagon-5}
1644 \group_end:
1645 }
1646 \prop_gput:Nnn \g__tilings_pentagon_lms_action_prop {Q} {
1647 \group_begin:
1648 \int_gincr:N \g__tilings_tile_int
1649 \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim/2}
1650 \pgftransformxshift{-\fp_to_dim:N \l__tilings_tmpa_fp}
1651 \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim*tand(54)/2}
1652 \pgftransformyshift{-\fp_to_dim:N \l__tilings_tmpa_fp}
1653 \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim/(1cm)}
1654 \pgftransformscale{\fp_use:N \l__tilings_tmpa_fp}
1655 \tl_set:Nx \l__tilings_tmpe_tl
1656 {
1657 {\int_use:N \g__tilings_tile_int}
1658 {\int_use:N \g__tilings_tiles_int}
1659 }
1660 \UseTile[
1661 every~ tile/.try,
1662 every~ pentagon/.try,

```

```

1663     every~
1664     \seq_item:Nn \l__tilings_pentagon_parity_seq
1665     {\l__tilings_pentagon_parity_int}
1666     \space pentagon/.try,
1667     every~ pentagon~ 3/.try,
1668     tile~ \int_use:N \g__tilings_tile_int/.try,
1669     tile/.try/.expand~ once=\l__tilings_tmpc_tl
1670 ]{pentagon-3}
1671 \group_end:
1672 }

1673 \prop_gput:Nnn \g__tilings_pentagon_lms_action_prop {R} {
1674   \group_begin:
1675   \int_gincr:N \g__tilings_tile_int
1676   \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim/2}
1677   \pgftransformxshift{-\fp_to_dim:N \l__tilings_tmpa_fp}
1678   \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim*tand(54)/2}
1679   \pgftransformyshift{-\fp_to_dim:N \l__tilings_tmpa_fp}
1680   \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim/(1cm)}
1681   \pgftransformscale{\fp_use:N \l__tilings_tmpa_fp}
1682   \tl_set:Nx \l__tilings_tmpc_tl
1683   {
1684     {\int_use:N \g__tilings_tile_int}
1685     {\int_use:N \g__tilings_tiles_int}
1686   }
1687   \UseTile[
1688     every~ tile/.try,
1689     every~ pentagon/.try,
1690     every~
1691     \seq_item:Nn \l__tilings_pentagon_parity_seq
1692     {\l__tilings_pentagon_parity_int}
1693     \space pentagon/.try,
1694     every~ pentagon~ 2/.try,
1695     tile~ \int_use:N \g__tilings_tile_int/.try,
1696     tile/.try/.expand~ once=\l__tilings_tmpc_tl
1697 ]{pentagon-2}
1698 \group_end:
1699 }

1700 \prop_gput:Nnn \g__tilings_pentagon_lms_action_prop {G} {
1701   \group_begin:
1702   \int_gincr:N \g__tilings_tile_int
1703   % \pgftransformrotate{198}
1704   \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim*cosd(72)}
1705   \pgftransformxshift{\fp_to_dim:N \l__tilings_tmpa_fp}
1706   \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim*tand(54)*cosd(72)}
1707   \pgftransformyshift{\fp_to_dim:N \l__tilings_tmpa_fp}
1708   \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim/(1cm)}
1709   \pgftransformscale{\fp_use:N \l__tilings_tmpa_fp}
1710   \tl_set:Nx \l__tilings_tmpc_tl
1711   {
1712     {\int_use:N \g__tilings_tile_int}
1713     {\int_use:N \g__tilings_tiles_int}
1714   }
1715   \UseTile[
1716     every~ tile/.try,
1717     every~ pentagram/.try,
1718     tile~ \int_use:N \g__tilings_tile_int/.try,
1719     tile/.try/.expand~ once=\l__tilings_tmpc_tl
1720 ]{pentagram}
1721 \group_end:
1722 }

```



```

1723 \prop_gput:Nnn \g__tilings_pentagon_lms_action_prop {B} {
1724   \group_begin:
1725   \int_gincr:N \g__tilings_tile_int
1726   % \pgftransformrotate{198}
1727   \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim*cosd(72)}
1728   \pgftransformxshift{\fp_to_dim:N \l__tilings_tmpa_fp}
1729   \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim*tand(54)*cosd(72)}
1730   \pgftransformyshift{\fp_to_dim:N \l__tilings_tmpa_fp}
1731   \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim/(1cm)}
1732   \pgftransformscale{\fp_use:N \l__tilings_tmpa_fp}
1733   \tl_set:Nx \l__tilings_tmpc_tl
1734   {
1735     {\int_use:N \g__tilings_tile_int}
1736     {\int_use:N \g__tilings_tiles_int}
1737   }
1738   \UseTile[
1739     every~ tile/.try,
1740     every~ boat/.try,
1741     tile~ \int_use:N \g__tilings_tile_int/.try,
1742     tile/.try/.expand~ once=\l__tilings_tmpc_tl
1743   ]{boat}
1744   \group_end:
1745 }

1746 \prop_gput:Nnn \g__tilings_pentagon_lms_action_prop {D} {
1747   \group_begin:
1748   \int_gincr:N \g__tilings_tile_int
1749   \pgftransformrotate{90}
1750   \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim*cosd(18)}
1751   \pgftransformxshift{-\fp_to_dim:N \l__tilings_tmpa_fp}
1752   \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim/(1cm)}
1753   \pgftransformscale{\fp_use:N \l__tilings_tmpa_fp}
1754   \tl_set:Nx \l__tilings_tmpc_tl
1755   {
1756     {\int_use:N \g__tilings_tile_int}
1757     {\int_use:N \g__tilings_tiles_int}
1758   }
1759   \UseTile[
1760     every~ tile/.try,
1761     every~ diamond/.try,
1762     tile~ \int_use:N \g__tilings_tile_int/.try,
1763     tile/.try/.expand~ once=\l__tilings_tmpc_tl
1764   ]{diamond}
1765   \group_end:
1766 }

1767 \ExplSyntaxOff

1768 \RequirePackage{tikz}
1769 \usetikzlibrary{tilings}
1770 \ProvidesFile {tikzlibrarytilings.polykite.code.tex}
1771 [2023/06/01 v2.0 TikZ pics for Aperiodical Polykite tiles]

```

**PolykiteTile** Define one of the family of polykite tiles. Needs a name and length parameters. A star option switches the side labels to enable the hat-turtle pairing of a tile with its “opposite”.

```

1772 \DeclareDocumentCommand \DefinePolykiteTile {s m m m}
1773 {
1774   \IfBooleanTF {#1}
1775   {
1776     \DefineTile {#2} {2 2 1 1 2 2 1 1 1 1 2 2 1 1}
1777   }
1778   {
1779     \DefineTile {#2} {1 1 2 2 1 1 2 2 2 2 1 1 2 2}
1780   }
1781   {
1782     {0 : #4}
1783     ++{90 : #3}
1784     ++{150 : #3}
1785     ++{240 : #4}
1786     ++{180 : #4}
1787     ++{-90 : #3}
1788     ++{210 : #3}
1789     ++{-60 : #4}
1790     ++{0 : #4}
1791     ++{0 : #4}
1792     ++{60 : #4}
1793     ++{-30 : #3}
1794     ++{30 : #3}
1795     ++{120 : #4}
1796   }
1797 }

```

Some predefined tiles. The aperiodical hat and turtle can be swapped in for each other as they use the same edge definitions. The spectral hat and turtle are designed to be used in the same diagram. Technically, the spectral and aperiodical hats are the same, but I figured it better to have two names for the two uses. The spectre uses the alternating edge scheme.

- Aperiodical and Spectral Hat.

```

1798   \DefinePolykiteTile{aperiodical hat}{sqrt(3)/2}{1/2}
1799   \DefinePolykiteTile{spectral hat}{sqrt(3)/2}{1/2}

```

- Aperiodical and Spectral Turtles.

```

1800   \DefinePolykiteTile{aperiodical turtle}{1/2}{sqrt(3)/2}
1801   \DefinePolykiteTile*{spectral turtle}{1/2}{sqrt(3)/2}

```

- Aperiodical Spectre.

```

1802   \DefineTile {spectre} {a A a A a A a A a A a A}
1803   {
1804     {0 : 1}
1805     ++{90 : 1}
1806     ++{150 : 1}
1807     ++{240 : 1}
1808     ++{180 : 1}
1809     ++{-90 : 1}
1810     ++{210 : 1}
1811     ++{-60 : 1}
1812     ++{0 : 1}
1813     ++{0 : 1}
1814     ++{60 : 1}
1815     ++{-30 : 1}
1816     ++{30 : 1}

```

```

1817     ++{120 : 1}
1818   }

```

- Meta Clusters

The mapping between the notation in the <https://arxiv.org/abs/2303.10798> is:

$$\begin{aligned}
 A^+, A^- &\mapsto a, A \\
 B^+, B^- &\mapsto b, B \\
 F^+, F^- &\mapsto c, C \\
 X^+, X^- &\mapsto d, D \\
 L &\mapsto 1
 \end{aligned}$$

```

1819 \DefineTile{meta cluster T}{A A b}
1820 {
1821 { -1.5 , -sqrt(3)/2 }
1822 { 1.5 , -sqrt(3)/2 }
1823 { 0 , sqrt(3) }
1824 }
1825 \DefineTile{meta cluster P}{1 D d A 1 D d b}
1826 {
1827 { -2.5, sqrt(3)/2 }
1828 { -2, 0 }
1829 { -1.5, -sqrt(3)/2 }
1830 { -0.5 , -sqrt(3)/2 }
1831 { 2.5 , -sqrt(3)/2 }
1832 { 2 , 0 }
1833 { 1.5 , sqrt(3)/2 }
1834 { .5 , sqrt(3)/2 }
1835 }
1836 \DefineTile{meta cluster F}{1 D d 1 D c C d b}
1837 {
1838 { -2.5, sqrt(3)/2 }
1839 { -2, 0 }
1840 { -1.5, -sqrt(3)/2 }
1841 { -0.5 , -sqrt(3)/2 }
1842 { .5 , -sqrt(3)/2 }
1843 { 1.5 , -sqrt(3)/2 }
1844 { 2 , 0 }
1845 { 1.5 , sqrt(3)/2 }
1846 { .5 , sqrt(3)/2 }
1847 }
1848 \DefineTile{meta cluster H}{B D d B D d a D d}
1849 {
1850 { -2, -sqrt(3) }
1851 { 1, -sqrt(3) }
1852 { 2, -sqrt(3) }
1853 { 2.5, -sqrt(3)/2 }
1854 { 1, sqrt(3) }
1855 { .5 , 3*sqrt(3)/2 }
1856 { -.5 , 3*sqrt(3)/2 }
1857 { -2, 0 }
1858 { -2.5, -sqrt(3)/2 }
1859 }

```

- Super Clusters

```

1860 \DefineTile{super cluster T}{A A b}

```

```

1861 {
1862 { -30 : 3 * (1 + sqrt(5))/2 / sqrt(3) }
1863 { 90 : 3 * (1 + sqrt(5))/2 / sqrt(3) }
1864 { 210 : 3 * (1 + sqrt(5))/2 / sqrt(3) }
1865 }
1866 % 1 + 3phi, 1 + 2phi
1867 \DefineTile{super cluster P}{1 D d A 1 D d b}
1868 {
1869 { - 1.75 - sqrt(5), (sqrt(5)/2 + 1) * sqrt(3)/2 }
1870 ++{ -60 : 1 + sqrt(5) }
1871 ++{ -60 : 1}
1872 ++{ 1, 0}
1873 ++{ 3*(1 + sqrt(5))/2, 0 }
1874 ++{ 120 : 1 + sqrt(5) }
1875 ++{ 120 : 1 }
1876 ++{ -1, 0 }
1877 }
1878 \DefineTile{super cluster F}{1 D d 1 D c C d b}
1879 {
1880 { - 1.75 - sqrt(5), (sqrt(5)/2 + 1) * sqrt(3)/2 }
1881 ++{ -60 : 1 + sqrt(5) }
1882 ++{ -60 : 1}
1883 ++{ 1, 0}
1884 ++{ 1 + sqrt(5), 0 }
1885 ++{ 1, 0 }
1886 ++{ -0.75 + (2 + sqrt(5)) * sqrt(3)/4 * sqrt(3)/3,
1887 (2 + sqrt(5)) * sqrt(3)/4 + 3/4 * sqrt(3)/3 }
1888 ++{ -1.5, (2 + sqrt(5)) * sqrt(3)/2 }
1889 ++{ -1, 0 }
1890 }
1891 \DefineTile{super cluster H}{B D d B D d a D d}
1892 {
1893 {1.75 + 3*sqrt(5)/4, -(1 + sqrt(5))*sqrt(3)/4}
1894 ++{120 : 3*(1+sqrt(5))/2 }
1895 ++{120 : 1 }
1896 ++{-1,0}
1897 ++{240 : 3*(1+sqrt(5))/2}
1898 ++{240 : 1}
1899 ++{300 : 1}
1900 ++{ 3*(1+sqrt(5))/2, 0 }
1901 ++{1, 0}
1902 ++{60 : 1}
1903 }

```

- Subclusters

```

1904 \DefineTile{subcluster H}{B B a}
1905 {
1906 { 0, 0 }
1907 { 3, 0 }
1908 { 60 : 3 }
1909 }
1910 \DefineTile{subcluster T}{A A b}
1911 {
1912 { 0, 0 }
1913 { 3, 0 }
1914 { 60 : 3 }
1915 }
1916 \DefineTile{subcluster P}{ 1 A 1 b }
1917 {
1918 { 0, 0 }

```

```

1919     { 1, 0 }
1920     { 4, 0 }
1921     { 3, 0 }
1922   }
1923   \DefineTile{subcluster F}{ 1 1 f F b }
1924   {
1925     { 0, 0 }
1926     { 1, 0 }
1927     +{ 60 : 1 }
1928     { 2, 0 }
1929     { 3, 0 }
1930   }

```

The P and F subclusters have no area, so clipping against them is not helpful.

```

1931 \tikzset{
1932   no clip/.code={%
1933     \tikz@addmode{\tikz@mode@clipfalse}%
1934   },
1935   every subcluster P clip/.style={no clip},
1936   every subcluster F clip/.style={no clip},
1937 }

1938 \BakeTile {aperiodical hat}
1939 \BakeTile {aperiodical turtle}
1940 \BakeTile {spectral hat}
1941 \BakeTile {spectral turtle}
1942 \BakeTile {spectre}
1943 \BakeTile {meta cluster T}
1944 \BakeTile {meta cluster P}
1945 \BakeTile {meta cluster F}
1946 \BakeTile {meta cluster H}
1947 \BakeTile {super cluster T}
1948 \BakeTile {super cluster P}
1949 \BakeTile {super cluster F}
1950 \BakeTile {super cluster H}

```

The subclusters are deformed by default.

```

1951 \ExplSyntaxOn
1952 \clist_map_inline:nn {a,A,b,B,f,F}
1953 {
1954   \tl_new:c {g__tilings_side_polykite_#1_tl}
1955   \tl_if_exist:cF {g__tilings_side_#1_tl}
1956   {
1957     \tl_new:c {g__tilings_side_#1_tl}
1958   }
1959 }
1960
1961 \tl_gset:cn {g__tilings_side_polykite_A_tl}
1962 {
1963   \pgfsyssoftpath@movetotoken {0pt}{-0.3333332942822268pt}
1964   \pgfsyssoftpath@linetotoken {0.0833331478405773pt}{-0.1889954840909892pt}
1965   \pgfsyssoftpath@linetotoken {0.3333332942822268pt}{-0.3333332942822268pt}
1966   \pgfsyssoftpath@linetotoken {0.583333235705567pt}{-0.1889954840909892pt}
1967   \pgfsyssoftpath@linetotoken {0.6666667057177732pt}{-0.3333332942822268pt}
1968   \pgfsyssoftpath@linetotoken {1pt}{-0.3333332942822268pt}
1969 }
1970 \tl_gset:cn {g__tilings_side_polykite_a_tl}
1971 {
1972   \pgfsyssoftpath@movetotoken {0pt}{0.33333pt}
1973   \pgfsyssoftpath@linetotoken {0.33333pt}{0.33333pt}
1974   \pgfsyssoftpath@linetotoken {0.41667pt}{0.189pt}

```

```

1975 \pgfsyssoftpath@linetotoken {0.66667pt}{0.33333pt}
1976 \pgfsyssoftpath@linetotoken {0.91667pt}{0.189pt}
1977 \pgfsyssoftpath@linetotoken {1pt}{0.33333pt}
1978 }
1979 \tl_gset:cn {g__tilings_side_polykite_B_tl}
1980 {
1981 \pgfsyssoftpath@movetotoken {0pt}{0pt}
1982 \pgfsyssoftpath@linetotoken {0.333332942822268pt}{0pt}
1983 \pgfsyssoftpath@linetotoken {0.416666592761237pt}{0.1443378101912376pt}
1984 \pgfsyssoftpath@linetotoken {0.666667057177732pt}{0pt}
1985 \pgfsyssoftpath@linetotoken {0.9166666178527835pt}{0.1443378101912376pt}
1986 \pgfsyssoftpath@linetotoken {1pt}{0pt}
1987 }
1988 \tl_gset:cn {g__tilings_side_polykite_b_tl}
1989 {
1990 \pgfsyssoftpath@movetotoken {0pt}{0pt}
1991 \pgfsyssoftpath@linetotoken {0.08333pt}{-0.14433pt}
1992 \pgfsyssoftpath@linetotoken {0.33333pt}{0pt}
1993 \pgfsyssoftpath@linetotoken {0.58333pt}{-0.14433pt}
1994 \pgfsyssoftpath@linetotoken {0.66667pt}{0pt}
1995 \pgfsyssoftpath@linetotoken {1pt}{0pt}
1996 }
1997 \tl_gset:cn {g__tilings_side_polykite_F_tl}
1998 {
1999 \pgfsyssoftpath@movetotoken {0pt}{-2.00000070292pt}
2000 \pgfsyssoftpath@linetotoken {0.74999982427pt}{-1.566987221617321pt}
2001 \pgfsyssoftpath@linetotoken {1pt}{-2.00000070292pt}
2002 }
2003 \tl_gset:cn {g__tilings_side_polykite_f_tl}
2004 {
2005 \pgfsyssoftpath@movetotoken {0pt}{2pt}
2006 \pgfsyssoftpath@linetotoken {0.25pt}{1.56699pt}
2007 \pgfsyssoftpath@linetotoken {1pt}{2pt}
2008 }
2009
2010 \clist_map_inline:nn {a,A,b,B,f,F}
2011 {
2012 \tl_gclear_new:c {g__tilings_side_backup_#1_tl}
2013 \tl_gset_eq:cc {g__tilings_side_backup_#1_tl} {g__tilings_side_#1_tl}
2014 \tl_gclear_new:c {g__tilings_side_#1_tl}
2015 \tl_gset_eq:cc {g__tilings_side_#1_tl}{g__tilings_side_polykite_#1_tl}
2016 }
2017
2018 \BakeTile{subcluster~ H}
2019 \BakeTile{subcluster~ T}
2020 \BakeTile{subcluster~ P}
2021 \BakeTile{subcluster~ F}
2022
2023 \clist_map_inline:nn {a,A,b,B,f,F}
2024 {
2025 \tl_gset_eq:cc {g__tilings_side_#1_tl} {g__tilings_side_backup_#1_tl}
2026 }

```

## 2.7 Lindenmayer System

These are the rules for generating the super cluster tilings with the Lindenmayer System procedure.

```

_cluster_tile:mn Useful auxiliary for placing a cluster tile from a particular set
2027 \cs_new_protected_nopar:Npn \__tilings_place_cluster_tile:mn #1#2
2028 {
2029 \group_begin:

```

```

2030 \int_gincr:N \g__tilings_tile_int
2031 \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim/(1cm)}
2032 \pgftransformscale{\fp_use:N \l__tilings_tmpa_fp}
2033 \tl_set:Nx \l__tilings_tmpc_tl
2034 {
2035   {\int_use:N \g__tilings_tile_int}
2036   {\int_use:N \g__tilings_tiles_int}
2037 }
2038 \UseTile[
2039   every~ tile/.try,
2040   every~ #1~#2/.try,
2041   tile~ \int_use:N \g__tilings_tile_int/.try,
2042   tile/.try/.expand~ once=\l__tilings_tmpc_tl
2043 ]{#1~ #2}
2044 \group_end:
2045 }
2046 \cs_generate_variant:Nn \__tilings_place_cluster_tile:nn {Vn}

(End definition for \__tilings_place_cluster_tile:nn.)

2047 \prop_new:N \g__tilings_supercluster_lms_rule_prop
2048 \prop_gput:Nnn \g__tilings_supercluster_lms_rule_prop {T}
2049 {
2050   [s H]
2051 }
2052 \prop_gput:Nnn \g__tilings_supercluster_lms_rule_prop {H}
2053 {
2054   [s {r{-60}} T]
2055   [s {x{\fp_to_decimal:n{1}}} {y{\fp_to_decimal:n{(1+2*\c__tilings_phi_fp)}}} H]
2056   [s
2057     {x{\fp_to_decimal:n{-2-3*\c__tilings_phi_fp}}}
2058     {y{\fp_to_decimal:n{-\c__tilings_phi_fp}}}
2059     H]
2060   [s
2061     {x{\fp_to_decimal:n{1+3*\c__tilings_phi_fp}}}
2062     {y{\fp_to_decimal:n{-1-\c__tilings_phi_fp}}}
2063     {r{-120}}
2064     H]
2065   [s
2066     {x{\fp_to_decimal:n{-1.5-3.5*\c__tilings_phi_fp}}}
2067     {y{\fp_to_decimal:n{2.5*\c__tilings_phi_fp+1.5}}}
2068     {r{-120}}
2069     P]
2070   [s
2071     {x{\fp_to_decimal:n{-1.5-2*\c__tilings_phi_fp}}}
2072     {y{\fp_to_decimal:n{-1.5-3*\c__tilings_phi_fp}}}
2073     {r{180}}
2074     P]
2075   [s
2076     {x{\fp_to_decimal:n{3+5.5*\c__tilings_phi_fp}}}
2077     {y{\fp_to_decimal:n{.5*\c__tilings_phi_fp}}}
2078     {r{120}}
2079     P]
2080   [s
2081     {x{\fp_to_decimal:n{-4.5-6.5*\c__tilings_phi_fp}}}
2082     {y{\fp_to_decimal:n{.5-.5*\c__tilings_phi_fp}}}
2083     {r{-120}}
2084     F]
2085   [s
2086     {x{\fp_to_decimal:n{1.5+4*\c__tilings_phi_fp}}}
2087     {y{\fp_to_decimal:n{-2.5-3*\c__tilings_phi_fp}}}
2088     F]

```

```

2089 [s
2090 {x{\fp_to_decimal:n{3+2.5*\c__tilings_phi_fp}}}
2091 {y{\fp_to_decimal:n{2+3.5*\c__tilings_phi_fp}}}
2092 {r{120}}
2093 F]
2094 }
2095 \prop_gput:Nnn \g__tilings_supercluster_lms_rule_prop {P}
2096 {
2097 [s {r{60}} P]
2098 [s
2099 {x{\fp_to_decimal:n{2.5+3.5*\c__tilings_phi_fp}}}
2100 {y{\fp_to_decimal:n{-0.5-0.5*\c__tilings_phi_fp}}}
2101 {r{-120}}
2102 H]
2103 [s
2104 {x{\fp_to_decimal:n{-2.5-3.5*\c__tilings_phi_fp}}}
2105 {y{\fp_to_decimal:n{0.5+0.5*\c__tilings_phi_fp}}}
2106 {r{180}}
2107 H]
2108 [s
2109 {x{\fp_to_decimal:n{4.5+6*\c__tilings_phi_fp}}}
2110 {y{\fp_to_decimal:n{0.5*\c__tilings_phi_fp}}}
2111 {r{120}}
2112 F]
2113 [s
2114 {x{\fp_to_decimal:n{-4.5-6*\c__tilings_phi_fp}}}
2115 {y{\fp_to_decimal:n{-0.5-\c__tilings_phi_fp}}}
2116 {r{-60}}
2117 F]
2118 }
2119 \prop_gput:Nnn \g__tilings_supercluster_lms_rule_prop {F}
2120 {
2121 [s {r{60}} P]
2122 [s
2123 {x{\fp_to_decimal:n{2.5+3.5*\c__tilings_phi_fp}}}
2124 {y{\fp_to_decimal:n{-0.5-0.5*\c__tilings_phi_fp}}}
2125 {r{-120}}
2126 H]
2127 [s
2128 {x{\fp_to_decimal:n{-2.5-3.5*\c__tilings_phi_fp}}}
2129 {y{\fp_to_decimal:n{0.5+0.5*\c__tilings_phi_fp}}}
2130 {r{180}}
2131 H]
2132 [s
2133 {x{\fp_to_decimal:n{4.5+6*\c__tilings_phi_fp}}}
2134 {y{\fp_to_decimal:n{0.5*\c__tilings_phi_fp}}}
2135 {r{120}}
2136 F]
2137 [s
2138 {x{\fp_to_decimal:n{-4.5-6*\c__tilings_phi_fp}}}
2139 {y{\fp_to_decimal:n{-0.5-\c__tilings_phi_fp}}}
2140 {r{-60}}
2141 F]
2142 [s
2143 {x{\fp_to_decimal:n{3+4.5*\c__tilings_phi_fp}}}
2144 {y{\fp_to_decimal:n{-2-2.5*\c__tilings_phi_fp}}}
2145 F]
2146 }
2147 \prop_gput:Nnn \g__tilings_drawables_lms_prop {supercluster} {HTPF}
2148 \fp_const:Nn \c__tilings_phi_fp {(1 + sqrt(5))/2}
2149 \prop_new:N \g__tilings_supercluster_lms_action_prop

```



```

2150
2151 \prop_gput:Nnn \g__tilings_supercluster_lms_action_prop {r}
2152 {
2153   \pgftransformrotate{\l__tilings_parameters_lms_tl}
2154 }
2155 \prop_gput:Nnn \g__tilings_supercluster_lms_action_prop {x}
2156 {
2157   \pgftransformxshift{
2158     \fp_to_dim:n
2159     {.5 * (\l__tilings_parameters_lms_tl) * \l__tilings_step_dim}
2160   }
2161 }
2162 \prop_gput:Nnn \g__tilings_supercluster_lms_action_prop {y}
2163 {
2164   \pgftransformyshift{
2165     \fp_to_dim:n
2166     {.5 * sqrt(3) * (\l__tilings_parameters_lms_tl) * \l__tilings_step_dim}
2167   }
2168 }
2169
2170 \prop_gput:Nnn \g__tilings_supercluster_lms_action_prop {s}
2171 {
2172   \fp_set:Nn \l__tilings_tmpa_fp {
2173     \l__tilings_step_dim
2174     /
2175     \c__tilings_phi_fp
2176     /
2177     \c__tilings_phi_fp
2178   }
2179   \dim_set:Nn \l__tilings_step_dim {\fp_to_dim:N \l__tilings_tmpa_fp}
2180 }
2181 \prop_gput:Nnn \g__tilings_supercluster_lms_action_prop {H}
2182 {
2183   \__tilings_place_cluster_tile:nn {super~ cluster}{H}
2184 }
2185 \prop_gput:Nnn \g__tilings_supercluster_lms_action_prop {T}
2186 {
2187   \__tilings_place_cluster_tile:nn {super~ cluster}{T}
2188 }
2189 \prop_gput:Nnn \g__tilings_supercluster_lms_action_prop {P}
2190 {
2191   \__tilings_place_cluster_tile:nn {super~ cluster}{P}
2192 }
2193 \prop_gput:Nnn \g__tilings_supercluster_lms_action_prop {F}
2194 {
2195   \__tilings_place_cluster_tile:nn {super~ cluster}{F}
2196 }

```

Parameters:

1. Cluster type (super cluster, meta cluster, subcluster)
2. This tile type (H, T, P, F)
3. This tile's name
4. Alignment tile's name
5. Edge to align along
6. Edge to align with

```

2197 \cs_new_protected_nopar:Npn \__tilings_place_cluster_tile_as_pic:nnnnnn #1#2#3#4#5#6
2198 {
2199   \group_begin:
2200   \int_gincr:N \g__tilings_tile_int
2201   \fp_set:Nn \l__tilings_tmpa_fp {\l__tilings_step_dim/(1cm)}
2202   \tl_set:Nx \l__tilings_tmpc_tl
2203   {
2204     {\int_use:N \g__tilings_tile_int}
2205     {\int_use:N \g__tilings_tiles_int}
2206   }
2207   \tl_clear:N \l__tilings_tmpa_tl
2208   \tl_put_right:Nn \l__tilings_tmpa_tl
2209   {
2210     \pic[
2211       every~ tile/.try,
2212       every~ #1~#2/.try,
2213     ]
2214     \tl_put_right:Nx \l__tilings_tmpa_tl
2215     {
2216       tile~ \int_use:N \g__tilings_tile_int/.try,
2217       tile/.try=\l__tilings_tmpc_tl,
2218       scale=\fp_use:N \l__tilings_tmpa_fp,
2219     }
2220     \tl_put_right:Nn \l__tilings_tmpa_tl
2221     {
2222       name=#3,
2223     }
2224     \tl_if_empty:nTF {#4}
2225     {
2226       \tl_put_right:Nn \l__tilings_tmpa_tl
2227       {
2228         first~ tile/.try,
2229       }
2230     }
2231     {
2232       \tl_put_right:Nn \l__tilings_tmpa_tl
2233       {
2234         align~ with=#4~along~#5
2235       }
2236       \tl_if_single:nF {#6}
2237       {
2238         \tl_put_right:Nx \l__tilings_tmpa_tl
2239         {
2240           \c_space_tl using~\tl_tail:n {#6}
2241         }
2242       }
2243       \tl_put_right:Nn \l__tilings_tmpa_tl {,}
2244     }
2245     \tl_put_right:Nn \l__tilings_tmpa_tl
2246     {
2247       #1~ #2
2248     ];
2249   }
2250   \tl_use:N \l__tilings_tmpa_tl
2251   \group_end:
2252 }
2253 \cs_generate_variant:Nn \__tilings_place_cluster_tile_as_pic:nnnnnn {
2254   Vnnnnn, VnVnnn, VnVVnn
2255 }
2256 \tikzset{
2257   cluster~ type/.initial=super~ cluster,

```

```

2258 first~ file/.style={transform~ shape}
2259 }
2260 \prop_new:N \g__tilings_cluster_lms_rule_prop

```

The first set of rules govern when the tile being replaced is a root tile, in which case one of the new tiles becomes the new root and all others are placed with respect to them.

It's convenient for code readability to have aliases for the labels for the parent and adjoining tiles, which are stored in the `\l_@@_parameters_lms_tl` token list.

```

2261 \cs_new_nopar:Npn \__tilings_tile_label:
2262 {
2263   \tl_item:Nn \l__tilings_parameters_lms_tl {1}
2264 }
2265 \cs_new_nopar:Npn \__tilings_adjoint_label:
2266 {
2267   \tl_item:Nn \l__tilings_parameters_lms_tl {2}
2268 }

```

A single  $T$  tile is replaced by a single  $H$  tile

```

2269 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {T}
2270 {
2271   [s {H{\__tilings_tile_label:0}{}}]
2272 }

```

An  $H$  tile is replaced by 10 tiles, consisting of a  $T$  tile and 3 each of  $H$ ,  $P$ , and  $F$ .

```

2273 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {H}
2274 {
2275   [s {r{-60}} {T{\__tilings_tile_label:0}{}}]
2276   [{ {HTa{A1}} {\__tilings_tile_label:1} {\__tilings_tile_label:0} }]
2277   [{ {HTa{A2}} {\__tilings_tile_label:2} {\__tilings_tile_label:0} }]
2278   [{ {HT{B1}b} {\__tilings_tile_label:3} {\__tilings_tile_label:0} }]
2279   [{ {PHb{B2}} {\__tilings_tile_label:4} {\__tilings_tile_label:1} }]
2280   [{ {PHb{B2}} {\__tilings_tile_label:5} {\__tilings_tile_label:2} }]
2281   [{ {PHAa} {\__tilings_tile_label:6} {\__tilings_tile_label:3} }]
2282   [{ {FHb{B1}} {\__tilings_tile_label:7} {\__tilings_tile_label:1} }]
2283   [{ {FHb{B1}} {\__tilings_tile_label:8} {\__tilings_tile_label:2} }]
2284   [{ {FHb{B2}} {\__tilings_tile_label:9} {\__tilings_tile_label:3} }]
2285 }

```

Lastly, the  $P$  and  $F$  tile substitutions.

```

2286 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {P}
2287 {
2288   [s {r{60}} {P{\__tilings_tile_label:0}{}}]
2289   [{ {HPaA} {\__tilings_tile_label:1} {\__tilings_tile_label:0} }]
2290   [{ {HP{B2}b} {\__tilings_tile_label:2} {\__tilings_tile_label:0} }]
2291   [{ {FHb{B2}} {\__tilings_tile_label:3} {\__tilings_tile_label:1} }]
2292   [{ {FHb{B1}} {\__tilings_tile_label:4} {\__tilings_tile_label:2} }]
2293 }
2294 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {F}
2295 {
2296   [s {r{60}} {P{\__tilings_tile_label:0}{}}]
2297   [{ {HPaA} {\__tilings_tile_label:1} {\__tilings_tile_label:0} }]
2298   [{ {HP{B2}b} {\__tilings_tile_label:2} {\__tilings_tile_label:0} }]
2299   [{ {FHb{B2}} {\__tilings_tile_label:3} {\__tilings_tile_label:1} }]
2300   [{ {FHb{B1}} {\__tilings_tile_label:4} {\__tilings_tile_label:2} }]
2301   [{ {FHb{B1}} {\__tilings_tile_label:5} {\__tilings_tile_label:1} }]
2302 }

```

The rest of the rules are for when the tile being replaced was itself positioned by aligning it with another tile. For these tiles, one of its edge tiles will be its root and positioned alongside one of the edge tiles of the replacement of the original tile's alignment tile. Then all the other tiles are positioned out from that root. The labelling has to be the same regardless of the order of drawing the tiles.

Not every edge pairing is necessary to generate a pattern as the edges that can be created are all between  $A^\pm$  edges and between  $B^\pm$  edges. However, to avoid errors in case they are part of the seed then for now we create blank substitution rules that will effectively remove any such rogue elements.

```

2303 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {TH{A1}a} {}
2304 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {TH{A2}a} {}
2305 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {THb{B1}} {}
2306 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {THb{B2}} {}
2307 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {HTa{A1}} {}
2308 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {HTa{A2}} {}
2309 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {HT{B1}b} {}
2310 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {HT{B2}b} {}
2311 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {HPaA} {}
2312 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {HP{B1}b} {}
2313 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {HP{B2}b} {}
2314 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {HF{B1}b} {}
2315 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {HF{B2}b} {}
2316 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {PHaA} {}
2317 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {PHb{B1}} {}
2318 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {PHb{B2}} {}
2319 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {PF{11}{11}} {}
2320 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {PF{12}{11}} {}
2321 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {PF{11}{12}} {}
2322 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {PF{12}{12}} {}
2323 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {FHB{B1}} {}
2324 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {FHB{B2}} {}
2325 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {FP{11}{11}} {}
2326 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {FP{12}{11}} {}
2327 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {FP{11}{12}} {}
2328 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {FP{12}{12}} {}
2329 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {FFfF} {}
2330 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {FFFf} {}
2331 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {FF{11}{11}} {}
2332 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {FF{12}{11}} {}
2333 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {FF{11}{12}} {}
2334 \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {FF{12}{12}} {}

```

To help create the rules then we start with some helper macros. Each of these creates the substitution rule for a tile given certain information about where the parent tile is positioned. Most of the substitution information consists of placing the tiles next to each other, so only the first tile needs to know about a tile from a different set. This makes it relatively easy to set up some templates for the substitution rules.

```

2335 \cs_new_nopar:cpn {__tilings_T{A1}_creator:nnnnn} #1#2#3#4#5
2336 {
2337   \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {T#1{A1}#2}
2338   {
2339     [{ {H#3{B1}#4} {\__tilings_tile_label:0} {\__tilings_adjoint_label:#5} ]}
2340   }
2341 }
2342 \cs_new_nopar:cpn {__tilings_T{A2}_creator:nnnnn} #1#2#3#4#5
2343 {
2344   \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {T#1{A2}#2}
2345   {
2346     [{ {H#3{B2}#4} {\__tilings_tile_label:0} {\__tilings_adjoint_label:#5} ]}
2347   }
2348 }
2349 \cs_new_nopar:cpn {__tilings_Tb_creator:nnnnn} #1#2#3#4#5
2350 {
2351   \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {T#1b#2}
2352   {
2353     [{ {H#3a#4} {\__tilings_tile_label:0} {\__tilings_adjoint_label:#5} ]}
2354   }
2355 }

```

```

2356
2357 \cs_new_nopar:cpn {__tilings_H{B1}_creator:nnnnn} #1#2#3#4#5
2358 {
2359   \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {H#1{B1}#2}
2360   {
2361     [{ {P#3A#4} {\__tilings_tile_label:4} {\__tilings_adjoint_label:#5} ]}
2362
2363     [{ {HP{B2}b} {\__tilings_tile_label:1} {\__tilings_tile_label:4} ]}
2364     [{ {TH{A1}a} {\__tilings_tile_label:0} {\__tilings_tile_label:1} ]}
2365
2366     [{ {HTa{A2}} {\__tilings_tile_label:2} {\__tilings_tile_label:0} ]}
2367     [{ {PHb{B2}} {\__tilings_tile_label:5} {\__tilings_tile_label:2} ]}
2368
2369     [{ {HT{B1}b} {\__tilings_tile_label:3} {\__tilings_tile_label:0} ]}
2370     [{ {PHaA} {\__tilings_tile_label:6} {\__tilings_tile_label:3} ]}
2371
2372     [{ {FHb{B1}} {\__tilings_tile_label:7} {\__tilings_tile_label:1} ]}
2373     [{ {FHb{B1}} {\__tilings_tile_label:8} {\__tilings_tile_label:2} ]}
2374     [{ {FHb{B2}} {\__tilings_tile_label:9} {\__tilings_tile_label:3} ]}
2375   }
2376 }
2377
2378 \cs_new_nopar:cpn {__tilings_H{B2}_creator:nnnnn} #1#2#3#4#5
2379 {
2380   \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {H#1{B2}#2}
2381   {
2382     [{ {P#3A#4} {\__tilings_tile_label:5} {\__tilings_adjoint_label:#5} ]}
2383
2384     [{ {HP{B2}b} {\__tilings_tile_label:2} {\__tilings_tile_label:5} ]}
2385     [{ {TH{A2}a} {\__tilings_tile_label:0} {\__tilings_tile_label:2} ]}
2386
2387     [{ {HTa{A1}} {\__tilings_tile_label:1} {\__tilings_tile_label:0} ]}
2388     [{ {PHb{B2}} {\__tilings_tile_label:4} {\__tilings_tile_label:1} ]}
2389
2390     [{ {HT{B1}b} {\__tilings_tile_label:3} {\__tilings_tile_label:0} ]}
2391     [{ {PHaA} {\__tilings_tile_label:6} {\__tilings_tile_label:3} ]}
2392
2393     [{ {FHb{B1}} {\__tilings_tile_label:7} {\__tilings_tile_label:1} ]}
2394     [{ {FHb{B1}} {\__tilings_tile_label:8} {\__tilings_tile_label:2} ]}
2395     [{ {FHb{B2}} {\__tilings_tile_label:9} {\__tilings_tile_label:3} ]}
2396   }
2397 }
2398
2399 \cs_new_nopar:cpn {__tilings_Ha_creator:nnnnn} #1#2#3#4#5
2400 {
2401   \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {H#1a#2}
2402   {
2403     [{ {P#3b#4} {\__tilings_tile_label:6} {\__tilings_adjoint_label:#5} ]}
2404
2405     [{ {HPaA} {\__tilings_tile_label:3} {\__tilings_tile_label:6} ]}
2406     [{ {THb{B1}} {\__tilings_tile_label:0} {\__tilings_tile_label:3} ]}
2407
2408     [{ {HTa{A1}} {\__tilings_tile_label:1} {\__tilings_tile_label:0} ]}
2409     [{ {PHb{B2}} {\__tilings_tile_label:4} {\__tilings_tile_label:1} ]}
2410
2411     [{ {HTa{A2}} {\__tilings_tile_label:2} {\__tilings_tile_label:0} ]}
2412     [{ {PHb{B2}} {\__tilings_tile_label:5} {\__tilings_tile_label:2} ]}
2413
2414     [{ {FHb{B1}} {\__tilings_tile_label:7} {\__tilings_tile_label:1} ]}
2415     [{ {FHb{B1}} {\__tilings_tile_label:8} {\__tilings_tile_label:2} ]}
2416     [{ {FHb{B2}} {\__tilings_tile_label:9} {\__tilings_tile_label:3} ]}

```

```

2417 }
2418 }
2419
2420 \cs_new_nopar:cpn {__tilings_PA_creator:nnnnn} #1#2#3#4#5
2421 {
2422   \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {P#1A#2}
2423   {
2424     [{ {H#3{B1}#4} {\__tilings_tile_label:1} {\__tilings_adjoint_label:#5} }]
2425     [{ {PHAa} {\__tilings_tile_label:0} {\__tilings_tile_label:1} }]
2426     [{ {HP{B2}b} {\__tilings_tile_label:2} {\__tilings_tile_label:0} }]
2427
2428     [{ {FHb{B2}} {\__tilings_tile_label:3} {\__tilings_tile_label:1} }]
2429     [{ {FHb{B1}} {\__tilings_tile_label:4} {\__tilings_tile_label:2} }]
2430   }
2431 }
2432
2433 \cs_new_nopar:cpn {__tilings_Pb_creator:nnnnn} #1#2#3#4#5
2434 {
2435   \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {P#1b#2}
2436   {
2437     [{ {H#3a#4} {\__tilings_tile_label:2} {\__tilings_adjoint_label:#5} }]
2438     [{ {PHb{B2}} {\__tilings_tile_label:0}{\__tilings_tile_label:2} }]
2439     [{ {HPaA} {\__tilings_tile_label:1} {\__tilings_tile_label:0} }]
2440
2441     [{ {FHb{B2}} {\__tilings_tile_label:3} {\__tilings_tile_label:1} }]
2442     [{ {FHb{B1}} {\__tilings_tile_label:4} {\__tilings_tile_label:2} }]
2443   }
2444 }
2445
2446 \cs_new_nopar:cpn {__tilings_Fb_creator:nnnnn} #1#2#3#4#5
2447 {
2448   \prop_gput:Nnn \g__tilings_cluster_lms_rule_prop {F#1b#2}
2449   {
2450     [{ {H#3a#4} {\__tilings_tile_label:2} {\__tilings_adjoint_label:#5} }]
2451     [{ {PHb{B2}} {\__tilings_tile_label:0}{\__tilings_tile_label:2} }]
2452     [{ {HPaA} {\__tilings_tile_label:1} {\__tilings_tile_label:0} }]
2453
2454     [{ {FHb{B2}} {\__tilings_tile_label:3} {\__tilings_tile_label:1} }]
2455     [{ {FHb{B1}} {\__tilings_tile_label:4} {\__tilings_tile_label:2} }]
2456     [{ {FHb{B1}} {\__tilings_tile_label:5} {\__tilings_tile_label:1} }]
2457   }
2458 }

```

Now that the creators are set up it is time to invoke them.

```

2459 \clist_map_inline:nn
2460 {
2461   TH{{A1}}a HP{{B1}}b06,
2462   TH{{A2}}a HP{{B2}}b06,
2463   THb{{B1}} HPaA04,
2464   THb{{B2}} HPaA05,
2465   PHAa HP{{B1}}b16,
2466   PHb{{B1}} HPaA24,
2467   PHb{{B2}} HPaA25,
2468   FHb{{B1}} HPaA24,
2469   FHb{{B2}} HPaA25,
2470 }
2471 {
2472   \tl_clear:N \l__tilings_tmpa_tl
2473   \tl_put_right:Nn \l__tilings_tmpa_tl { \use:c }
2474   \tl_put_right:Nx \l__tilings_tmpa_tl
2475   {
2476     {\__tilings_\tl_item:nn {#1}{2}\tl_item:nn {#1}{4} _creator:nnnnn}

```

```

2477     {\tl_item:nn {#1}{1}}{\tl_item:nn{#1}{3}}
2478     \tl_item:nn {#1}{5}\tl_item:nn{#1}{7}
2479     \tl_item:nn {#1}{9}
2480   }
2481   \tl_use:N \l__tilings_tmpa_tl
2482   \tl_clear:N \l__tilings_tmpa_tl
2483   \tl_put_right:Nn \l__tilings_tmpa_tl { \use:c }
2484   \tl_put_right:Nx \l__tilings_tmpa_tl
2485   {
2486     {__tilings_ \tl_item:nn {#1}{1}\tl_item:nn {#1}{3} _creator:nnnnn}
2487     {\tl_item:nn {#1}{2}}{\tl_item:nn{#1}{4}}
2488     \tl_item:nn {#1}{6}\tl_item:nn{#1}{8}
2489     \tl_item:nn {#1}{10}
2490   }
2491   \tl_use:N \l__tilings_tmpa_tl
2492 }
2493
2494 \prop_new:N \g__tilings_cluster_lms_action_prop
2495 \prop_gput:Nnn \g__tilings_drawables_lms_prop {cluster} {HTPF}
2496 \prop_gput:Nnn \g__tilings_cluster_lms_action_prop {r}
2497 {
2498   \pgftransformrotate{\l__tilings_parameters_lms_tl}
2499 }
2500 \prop_gput:Nnn \g__tilings_cluster_lms_action_prop {x}
2501 {
2502   \pgftransformxshift{
2503     \fp_to_dim:n
2504     {.5 * (\l__tilings_parameters_lms_tl) * \l__tilings_step_dim}
2505   }
2506 }
2507 \prop_gput:Nnn \g__tilings_cluster_lms_action_prop {y}
2508 {
2509   \pgftransformyshift{
2510     \fp_to_dim:n
2511     {.5 * sqrt(3) * (\l__tilings_parameters_lms_tl) * \l__tilings_step_dim}
2512   }
2513 }
2514
2515 \prop_gput:Nnn \g__tilings_cluster_lms_action_prop {s}
2516 {
2517   \fp_set:Nn \l__tilings_tmpa_fp {
2518     \l__tilings_step_dim
2519     /
2520     \c__tilings_phi_fp
2521     /
2522     \c__tilings_phi_fp
2523   }
2524   \dim_set:Nn \l__tilings_step_dim {\fp_to_dim:N \l__tilings_tmpa_fp}
2525 }

```

The first set of actions are for when this tile is the root so doesn't have a parent

```

2526 \clist_map_inline:nn {H,T,P,F}
2527 {
2528
2529   \prop_gput:Nnn \g__tilings_cluster_lms_action_prop {#1}
2530   {
2531     \__tilings_tikz_keys_get:Nn \l__tilings_tmpa_tl {cluster~type}
2532     \tl_set:Nx \l__tilings_tmpb_tl {\tl_item:Nn \l__tilings_parameters_lms_tl {1}}
2533     \__tilings_place_cluster_tile_as_pic:VnVnnn
2534     \l__tilings_tmpa_tl {#1} \l__tilings_tmpb_tl {}{}{}

```

```

2535 }
2536 }
    The second set is for when there is an adjoining edge
2537 \cs_new_protected_nopar:Npn \__tilings_place_cluster_tile_as_pic_aux:nnnn #1#2#3#4
2538 {
2539   \__tilings_tikz_keys_get:Nn \l__tilings_tmpa_tl {cluster~type}
2540   \tl_set:Nx \l__tilings_tmpb_tl {\__tilings_tile_label:}
2541   \tl_set:Nx \l__tilings_tmpc_tl {\__tilings_adjoint_label:}
2542   \__tilings_place_cluster_tile_as_pic:VnVVnn
2543   \l__tilings_tmpa_tl {#1} \l__tilings_tmpb_tl \l__tilings_tmpc_tl {#4}{#3}
2544 }
2545
2546 \clist_map_inline:mn {
2547   TH{A1}a,
2548   TH{A2}a,
2549   THb{B1},
2550   THb{B2},
2551   HTa{A1},
2552   HTa{A2},
2553   HT{B1}b,
2554   HT{B2}b,
2555   HPaA,
2556   HP{B1}b,
2557   HP{B2}b,
2558   HF{B1}b,
2559   HF{B2}b,
2560   PHAa,
2561   PHb{B1},
2562   PHb{B2},
2563   PF{11}{11},
2564   PF{12}{11},
2565   PF{11}{12},
2566   PF{12}{12},
2567   FHb{B1},
2568   FHb{B2},
2569   FP{11}{11},
2570   FP{12}{11},
2571   FP{11}{12},
2572   FP{12}{12},
2573   FFfF,
2574   FFFF,
2575   FF{11}{11},
2576   FF{12}{11},
2577   FF{11}{12},
2578   FF{12}{12}
2579 }
2580 {
2581   \prop_gput:Nnn \g__tilings_cluster_lms_action_prop {#1}
2582   {
2583     \__tilings_place_cluster_tile_as_pic_aux:nnnn #1
2584   }
2585 }
2586
2587 \ExplSyntaxOff
2588 \ProvidesFile {tikzlibrarypenrose.code.tex}
2589 [2023/06/01 v2.0 TikZ pics for Penrose tiles]
2590 \usetikzlibrary{tilings.penrose}

```

Backwards compatibility mode:

- \SetPenrosePath is \SetTilingPath



- \BakePenroseTile and \MakePenroseTile are \BakeTile
- \UsePenroseTile is \UseTile
- \PenroseDecomposition is \TilingDecomposition

```

2591 \ExplSyntaxOn
2592 \NewDocumentCommand \SetPenrosePath { m }
2593 {
2594   \__tilings_set_tiling_path:n {#1}
2595 }
2596 \NewDocumentCommand \BakePenroseTile {m}
2597 {
2598   \__tilings_bake_tile:n {#1}
2599 }
2600 \NewDocumentCommand \MakePenroseTile {m}
2601 {
2602   \__tilings_bake_tile:n {#1}
2603 }
2604 \NewDocumentCommand \UsePenroseTile {0{} m}
2605 {
2606   \__tilings_use_tile:nn {#1}{#2}
2607 }
2608 \NewDocumentCommand \PenroseDecomposition { 0{} m m m }
2609 {
2610   \__tilings_tiling_decomposition:nnnn {#1}{#2}{#3}{#4}
2611 }
2612 \ExplSyntaxOff
2613 \tikzset{
2614   save Penrose path/.forward to=/tikz/save tiling path,
2615   clone Penrose side path/.forward to=/tikz/clone tiling side path,
2616   spath/prefix/Penrose side/.forward to=/tikz/spath/prefix/tiling side,
2617   spath/suffix/Penrose side/.forward to=/tikz/spath/suffix/tiling side,
2618   clone Penrose tile path/.forward to=/tikz/clone tiling tile path,
2619   spath/prefix/Penrose tile/.forward to=/tikz/spath/prefix/tiling tile,
2620   spath/suffix/Penrose tile/.forward to=/tikz/spath/suffix/tiling tile,
2621   Penrose step/.forward to=/tikz/tiling step,
2622   every tile/.append style={
2623     every Penrose tile/.try
2624   },
2625   every tile clip/.append style={
2626     every Penrose tile clip/.try
2627   },
2628   every tile pic/.append style={
2629     every Penrose pic/.try
2630   },
2631   tile/.append style={
2632     Penrose tile #1/.try,
2633     Penrose tile/.try=#1
2634   }
2635 }

```