# Chapter 18
# The Game of Life Rules on Penrose Tilings: Still Life and Oscillators

Nick Owens and Susan Stepney

John Horton Conway's *Game of Life* [1, 4] is a simple two-dimensional, two state cellular automaton (CA), remarkable for its complex behaviour [1, 13]. That behaviour is known to be very sensitive to a change in the CA *rules*. Here we continue our investigations [7, 10, 11] into its sensitivity to changes in the *lattice*, by the use of an aperiodic Penrose tiling lattice [5, 12].

Section 18.1 describes Penrose tilings; Sect. 18.2 generalises the concepts of neighbourhood and outer totalistic CA rules (which include the Game of Life) to aperiodic lattices, and introduces a naming convention for Penrose Life oscillators. Section 18.3 presents various Penrose lattice still life configurations; Sects. 18.4–18.7 present various oscillators with periods from 2 to 15. Section 18.8 presents an algorithm to detect oscillators, and a means to classify them based on their underlying neighbourhood graph.

## 18.1 Penrose Tiling

### *18.1.1 Kites and Darts, and Rhombs*

Grünbaum & Shephard [6, chapter 10] provide a good introduction to aperiodic tilings, including Penrose tilings. The two variants of Penrose tiling we consider here are 'kites and darts', and 'rhombs'.

The kite and dart tile pair are shown in Fig. 18.1a; a large patch of kite and dart tiling is shown in Fig. 18.11. The thick and thin rhomb tile pair are shown in Fig. 18.1b; a large patch of rhomb tiling is shown in Fig. 18.13.

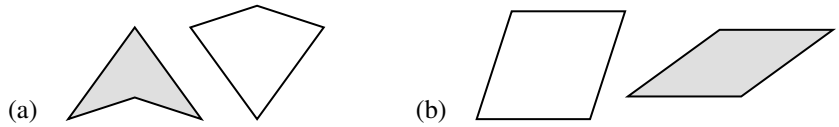


**Fig. 18.1** Penrose tiles: (a) the dart (grey) and kite (white) tiles: the long and short sides are in the ratio $\phi : 1$, where the golden ratio $\phi = (1+\sqrt{5})/2 = 2\cos(\pi/5)$; (b) the thick (white) and thin (grey) rhomb tiles

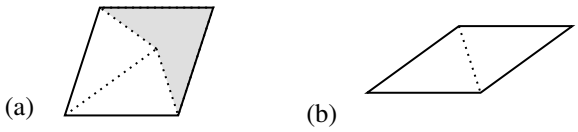


**Fig. 18.2** Relationship between rhomb tiles and kites and darts: (a) a thick rhomb comprises a dart and two half-kites (matching rules, later, forbid a dart and full kite from being joined in this way); (b) a thin rhomb comprises two half-kites

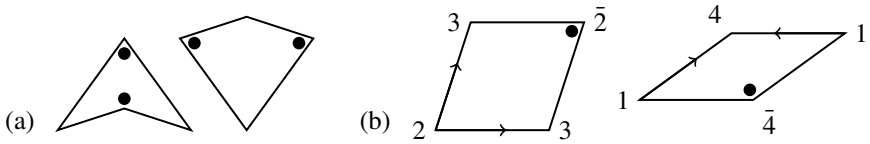


**Fig. 18.3** Matching rules: (a) kite and dart vertex markings; (b) rhomb vertex marking and edge orientations plus vertex angle numbering, where interior angles are $\pi/5$ times the vertex angle number (note that vertices labelled 2, and labelled 4, come in two kinds, due to the matching rules: these are distinguished by overbars)

### *18.1.2 Matching Rules*

The relationship of the rhomb tiles to the kite and dart tiles is shown in Fig. 18.2.

To avoid a kite and dart being joined to form a rhombus (Fig. 18.2a), which would allow a periodic tiling, there are additional 'matching rules' that force the tiling to be aperiodic: as well as edges of the same length being put together, certain vertices (given by the dots in Fig. 18.3a) must also be matched [5, 6].

To avoid rhomb tiles being used to form a periodic tiling, and force a periodic tiling, there are again additional 'matching rules': as well as edges of the same length being put together, the edge orientations (given by the arrows and dots in Fig. 18.3b) must also be matched [2].

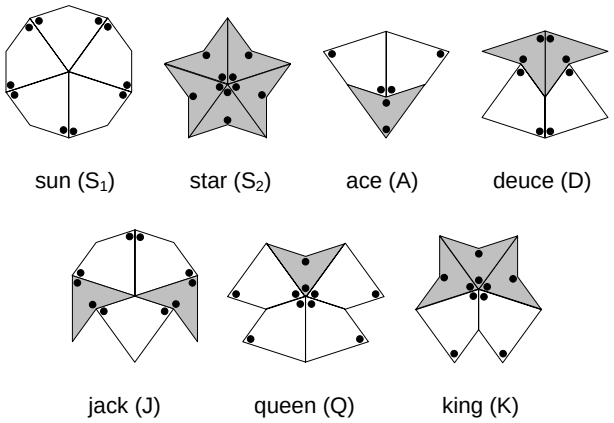


**Fig. 18.4** The seven valid vertex configurations of a kite and dart tiling [5]

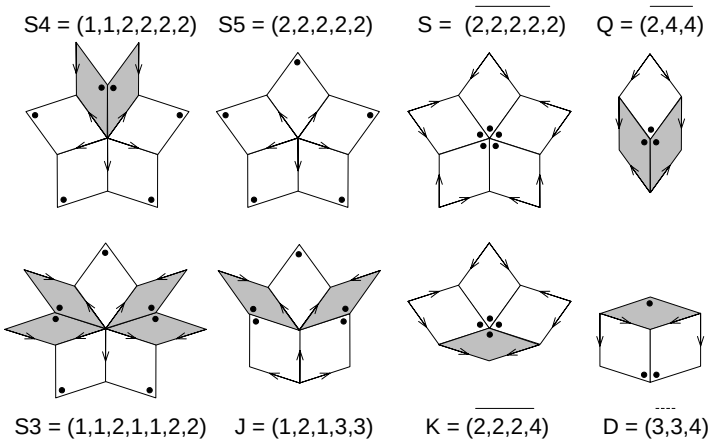


**Fig. 18.5** The eight valid vertex configurations of a rhomb tiling [2]

## *18.1.3 Valid Vertex Configurations*

There are many ways to put the tiles together, even with the restriction of the matching rules. However, in a true Penrose tiling (one that can be extended to infinity), not all of these configurations can exist.

There are only seven valid ways to surround any vertex in a kite and dart tiling [5] (Fig. 18.4).

There are only eight valid vertices in a rhomb tiling [2] (Fig. 18.5). The names of these vertices come from the names of the corresponding kite and dart vertices from which they can be derived [2]. Each vertex can be associated with a list of vertex angle numbers (after [14, fig.6.8], augmented here with overbars, Fig. 18.3b), corresponding to the vertex angles of the tiles forming the central vertex. These are

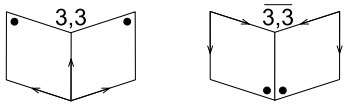


**Fig. 18.6** Disambiguating the 3,3 vertices: the two distinct ways a 3,3 vertex can appear in a valid rhomb vertex configuration (in the J and D, see Fig. 18.5). This is a *context dependent* marking [16]


**Fig. 18.7** Regular grid neighbourhoods: (a) the Moore neighbourhood, the eight cells with which the central cell shares a vertex; (b) the von Neumann neighbourhood, the four cells with which the central cell shares an edge

useful for determining how to complete forced vertices (see [10]). Note that there are two distinct occurrences of the 3,3 vertex configurations (in the J and D); see Fig. 18.6.

If a patch of tiling exhibits any other vertex configuration, it is not a true Penrose tiling: it will not be possible to extend it to infinity. We use these valid vertex configurations to analyse valid neighbourhood configurations later.

## 18.2 The Game of Life on a Penrose Tiling

### *18.2.1 Regular Game of Life Rules*

Classic cellular automata are defined on regular lattices. The update rule depends on the state of each cell and its neighbourhood (the surrounding cells)[1], and the structure of that neighbourhood is invariant: all places in the lattice look the same, and the update rule can be applied uniformly across the lattice.

In general, the update rule depends on the particular state of each separate neighbour. For *outer totalistic* CA rules, like the Game of Life (GoL), the next state of a cell depends only on its current state, and the total number of neighbourhood cells in certain states.

In GoL, the neighbourhood of each cell comprises the regular Moore neighbourhood (Fig. 18.7a), the eight cells with which it shares a vertex. Each cell has two states, 'dead' and 'alive'. If a cell is alive at time $t$, then it stays alive if and only if it has two or three live neighbours (otherwise it dies of 'loneliness' or 'overcrowd-

[1] The standard definition of CA 'neighbourhood' includes both the surrounding cells and the updating cell. Here we use slightly different terminology: by 'neighbourhood' we mean *only* the surrounding cells.

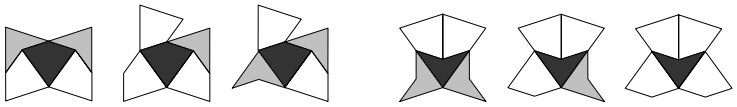

**Fig. 18.8** The generalised von Neumann neighbourhoods of a kite and dart Penrose tiling

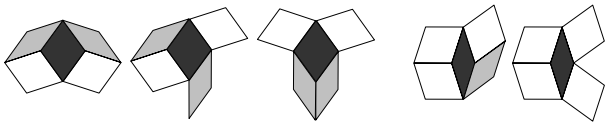

**Fig. 18.9** The generalised von Neumann neighbourhoods of a rhomb Penrose tiling

ing'). If a cell is dead at time $t$, then it becomes alive (is 'born') if and only if it has exactly three live neighbours.

For aperiodic lattices such as a Penrose tiling, the detailed structure of the neighbourhood varies at different locations in the lattice. Outer totalistic rules can be given an interpretation in these aperiodic tiling neighbourhoods.

## *18.2.2 Generalising the Neighbourhood and the Rules*

### 18.2.2.1 Generalised von Neumann Neighbourhood

The (Penrose) Game of Life rules use a (generalised) Moore neighbourhood. It is also convenient to define a generalised von Neumann neighbourhood, which we use later in the analysis of certain simple still lifes and oscillators.

Define the generalised von Neumann neighbourhood of a cell in a Penrose tiling to be all the cells with which it shares an edge (or, equivalently, two distinct vertices). Hence the size of the neighbourhood equals the number of edges of the central cell: four. Figures 18.8 and 18.9 show the distinct generalised von Neumann neighbourhoods which form valid vertices (established by exhaustive consideration of the valid vertex configurations, Figs. 18.4 and 18.5). Rotations and mirror images of these neighbourhoods are not considered to be distinct. de Bruijn [3] identifies the same rhomb neighbourhoods (but considers mirror images separately), and shows that a valid Penrose rhomb tiling can be constructed by considering just these neighbourhoods, without the need to use the rhomb matching rules of Fig. 18.3.

In the rectangular lattice none of the four von Neumann neighbourhood cells themselves share an edge. So if $A$ is a neighbour of $B$, and $B$ is a neighbour of $C$, then $A$ is *not* a neighbour of $C$: neighbouring von Neumann neighbourhoods do not overlap (recall that we do not treat the central site as a member of the neighbourhood here). In the Penrose lattice, this is no longer the case: cells in a generalised von Neumann neighbourhood can share an edge, so neighbouring generalised von

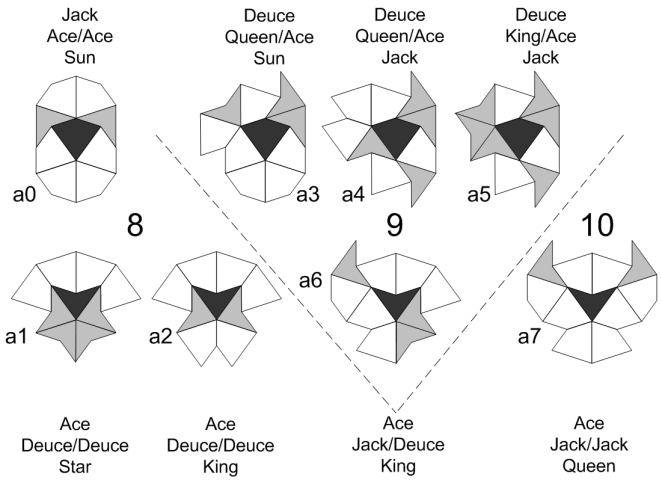


**Fig. 18.10** The generalised Moore neighbourhoods on a kite and dart Penrose tiling, with neighbourhood sizes, and the types of each vertex. Note that an ace vertex appears in every neighbourhood

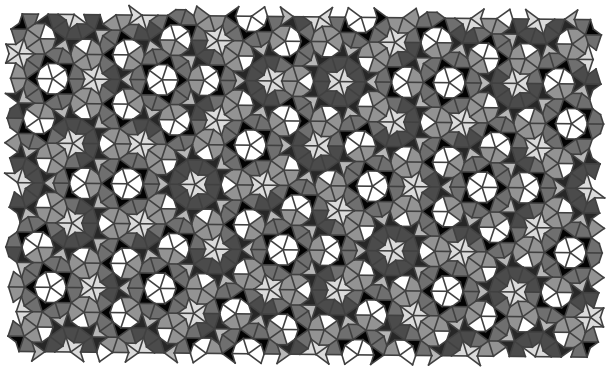

**Fig. 18.11** A kite and dart tiling shaded by neighbourhood type. The neighbourhood shading is uniformly distributed between white and black such that $a_0$ is white and $a_7$ black

Neumann neighbourhoods can overlap. This may affect the communication paths through the Penrose CA.

#### 18.2.2.2 Generalised Moore Neighbourhood

Define the generalised Moore neighbourhood of a cell in a Penrose tiling to be all the cells with which it shares a vertex.

Not only do cells have irregular shaped neighbourhoods, with the generalised Moore neighbourhood, not all cells have the same number of neighbours. The range of neighbourhood sizes and configurations is limited. Figure 18.10 shows the eight

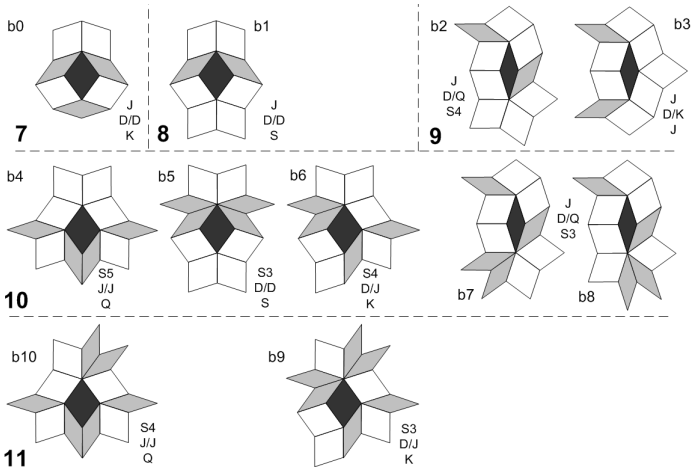


**Fig. 18.12** The generalised Moore neighbourhoods on a rhomb Penrose tiling, with neighbourhood sizes, and the types of each vertex. Note that b7 and b8 have the same types of vertices, but in different orientations: we call b7 the "ortho" form, and b8 the "para" form. Not that the 3 angle in the fat rhombs is always a J or D vertex. Note that one of the 1 angles in the thin rhomb is always a J vertex

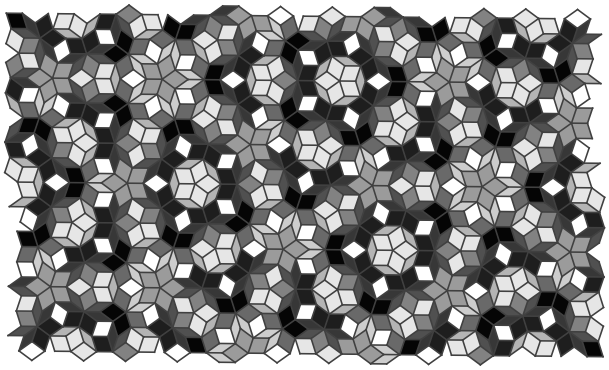

**Fig. 18.13** A rhomb tiling shaded by neighbourhood type. The neighbourhood shading is uniformly distributed between white and black such that $b_0$ is white and $b_{10}$ black. Socolar & Steinhardt [16] show that the ratio of thick to thin rhombs in a Penrose tiling is the golden ratio $\phi : 1$

valid neighbourhood configurations in a kite and dart tiling: there are no other valid ways to surround a kite or a dart (established by exhaustive consideration of the valid kite and dart vertex configurations, Fig. 18.4). So there is one neighbourhood configuration of size 8 around a kite, and two around a dart; three of size 9 around a kite, and one around a dart; and one of size 10, around a dart ([7] incorrectly states that kite and dart tilings have neighbourhoods of size 8 and 9 only). Figure 18.11 shows an area of kite and dart tilings with colouring to highlight the size of cells' neighbourhoods.

Similarly, Fig. 18.12 shows the valid neighbourhood configurations in a rhomb tiling (established by exhaustive consideration of the valid rhomb vertex configurations, Fig. 18.5). There is a larger range of distinct neighbourhood configurations for rhomb tilings. Figure 18.13 show an area of rhomb tilings with colouring to highlight the size of cells' neighbourhoods.

As can be seen from Figs. 18.11 and 18.13, not all sizes of neighbourhoods appear with the same frequency. [11] gives the distribution of neighbourhood sizes in a kite and dart tiling and in a rhomb tiling.

#### 18.2.2.3 Penrose Life Rules

Using our definition of the generalised Moore neighbourhood, the definition of the Game of Life as given in Sect. 18.2, in terms of the number of live and dead neighbours, can be used unchanged on a Penrose lattice.

Some early investigations are reported in [7]; further investigations are reported in [10, 11]. We find that the Game of Life rules on Penrose tilings still has complex behaviour. Both kinds of tiling need to be investigated, because we find that the Game of Life on the rhomb tiling is statistically significantly different from that on the kite and dart tiling: it has longer lifetimes to quiescence (the final periodic activity state), lower ash densities (density of 'live' cells at quiescence), higher soup growth (eventual extent of an initial random patch), significantly fewer oscillators in the ash, and lower ash period.

Here we continue our Penrose Life investigations, into the variety of oscillators supported by the tilings.

### *18.2.3 Identifying Oscillators*

#### 18.2.3.1 An Identification Code

GoL *patterns* are defined in terms of their 'live' cells. An *oscillator* is a pattern that recurs after a given number of generations, called the *period* of the oscillator. The Life Lexicon [15] defines the *rotor* to be "the cells of an oscillator that change state" and the *stator* to be "the cells of an oscillator that are always on". It also states that it is "easy to see that any rotor cell must be adjacent to another rotor cell"[2].

Following the regular Game of Life, we give some of the oscillators fanciful names, loosely based on their static appearance at one or more timesteps, or their

[2] *Proof*: Clearly a rotor cell must be adjacent to some other cell in the oscillator, else it would be dead. Consider an "off" rotor cell adjacent to only stator cells. Since it is a rotor cell, at some point it changes state to "on", at which time it has three "on" neighbours. Since these neighbours are stator cells by assumption, it would always have three "on" neighbours, so could no turn "off", and so could not be a rotor cell. We have a contradiction, so such a rotor cell cannot exist.

dynamic appearance as they oscillate. We generally name a Penrose oscillator after the isomorphic regular Life oscillator, if one exists.

In regular Life, oscillators are considered to be the same only if they have the same 2D pattern of cells, up to a rotation. But with the Penrose grid, there are patterns that can look superficially different (using a kite rather than a dart, or a thick rather than a thin rhomb), but have the same underlying structure. To help in their identification we give all oscillators a short *code*, for example, r-p2-8-6. The code has the following four-part form:

$$[\ l|kd|r\ -\ ]\ p\text{nn - xx [ - yy ]} \tag{18.1}$$

The first part, $l|kd|r$, identifies the *tiling*: whether we are talking about a regular life tiling $l$, a kite and dart tiling $kd$, or a rhomb tiling $r$. If the code is being used to refer to a pattern on all tilings, or if the tiling is clear from context, this part may be omitted.

The second part, $p$nn, identifies the *period* of the oscillator. For example, $p1$ means a still life.

The third part, xx, identifies the total number of cells involved in the oscillator. These are the cells that are "on", at some timestep. (For still lifes, this is just the number of cells.)

The fourth part, yy, is the minimum number of cells involved in the oscillator. It is determined by looking at the number of cells "on" in each timestep, and taking the minimum of these. (For still lifes, this is the same as the total, and so is omitted.) This follow the classification used in regular Life catalogues such as [9].

So, for example: p1-4 refers to a still life with four cells, that exists on all tilings (it may be either a *block* or a *tub*, see later); r-p2-8-6 refers to a rhomb period 2 oscillator, with a total of 8 cells live over its period, and a minimum of 6 live cells in one timestep (a *marcher*, see later).

Note that the code is not sufficient to uniquely identify an oscillator. Two clearly different oscillators may share a code; in particular, there are many $p1$ still lifes that have the same number of cells, but very different shapes. The third part of the code gives the number of nodes in the underlying "oscillator graph" (§18.8.2), which exposes this structure (the underlying topology of cell connections). We use this graph in addition to the code to identify "essentially similar", or *isomorphic*, oscillators: these have the same code *and* the same graph.

This scheme also allows us to say that oscillators on different tilings are nevertheless isomorphic. So, some regular Life oscillators can also exist on Penrose tilings, and some oscillators can exist on both forms of Penrose tiling.

#### 18.2.3.2 Variant Forms

Despite the Penrose tiling being aperiodic, it has much underlying structure. Some isomorphic oscillators are due to this structure, and can be systematically constructed from regularities in the underlying tiling.

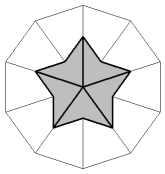

**Fig. 18.14** The "Complete star" ($S_C$), found by extending all the "forced" vertices of the Star

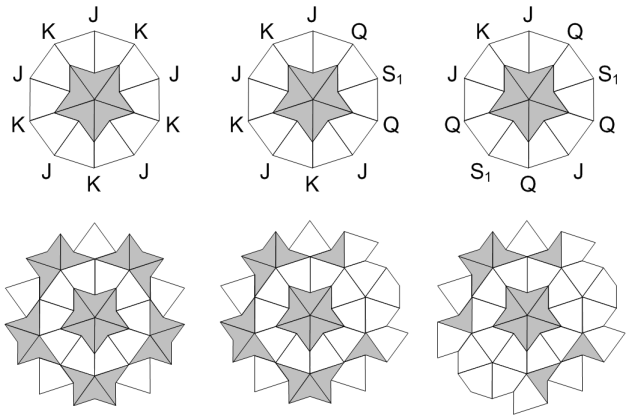


**Fig. 18.15** The three ways to extend the $S_C$ tiling. The top row labels the perimeter vertices with the vertices used to extend the tiling; the bottom row shows the resulting tilings

The underlying structure is generally captured in terms of "empires", or tiles forced to be in certain positions given the existence of particular other tiles (see, for example, [6, fig.10.6.6]). A *forced vertex* is one that can be completed in only one way to give a valid vertex. Completing the tiling around a forced vertex may result in new forced vertices; that a vertex is forced may be a result of constraints imposed by several surrounding vertices.

One class of variants is given by the different ways that a completed kite and dart Star vertex can be extended, described here. Consider a patch of tiling comprising a Star $S_2$ (Fig. 18.4), formed from five inward-pointing darts. Each of the Star's inward pointing vertices is forced to be an ace. Filling in these vertices results in what we call the "Complete Star", or $S_C$ (Fig. 18.14).

The Complete Star $S_C$ has no further forced vertices. However, there are still constraints on how further tiles can be attached to produce valid tilings. There are precisely three different ways to extend $S_C$, shown in Fig. 18.15. The different extensions can support variant forms of a given oscillator: three forms of the kd-p1-15 ring (Fig. 18.36), the symmetric and asymmetric forms of kd-p2-12-9, the *fast shuffler* (Fig. 18.48), the symmetric and asymmetric forms of kd-p4-14-6, the *bat* (Fig. 18.71), and three forms of kd-p15-40-8, the *dancer* (Fig. 18.71).

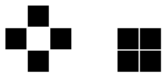

**Fig. 18.16** The regular Life tub and block still lifes, l-p1-4

## 18.3 Still Life

A *still life* is a pattern that remains constant: it can be thought of as a period 1 ($p1$) oscillator, or an oscillator that has no rotor component. (Strictly, a still life is a minimal such pattern, where no subset of its cells can be removed and leave a still life.)

Here we give a preliminary catalogue of Penrose still lifes. These have been discovered by a combination of systematic construction and random search: systematic construction of the small still lifes possible around certain vertices and neighbourhoods; an examination of ash contents from multiple runs of the GoL rules on random initial conditions (300,000 runs over a range of initial conditions, for each tiling [11]); and constructions of large still lifes that extend the structure of smaller ones. Because of the diversity of tiling patterns, there is no guarantee that the catalogue is exhaustive, particularly for larger numbers of tiles. Some of these still lifes clearly have commonalities, and some we classify as isomorphic. We provide a definition of oscillator isomorphism in §18.8.2.

### *18.3.1 Blocks and Tubs*

The smallest still lifes in regular Life have four cells (see [9] for an enumeration of all small still lifes); they are the *tub* (or *diamond*) and the *block* still lifes (Fig. 18.16). A tub is four "on" cells forming a chain around an "off" cell; the "on" cells are precisely the von Neumann neighbourhood of the surrounded "off" cell. A block is four "on" cells sharing a vertex.

These have isomorphic forms in Penrose Life, occurring in a variety of forms due to the more variable grid.

#### 18.3.1.1 Tubs

The Penrose Life isomorphic forms of the tub are formed from the generalised von Neumann neighbourhoods (Figs. 18.8 and 18.9). All the resulting *tubs* are shown in Fig. 18.17.

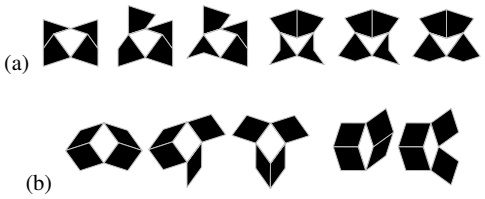


**Fig. 18.17** All the *tub* still lifes, p1-4: a) kite and dart tubs; b) rhomb tubs

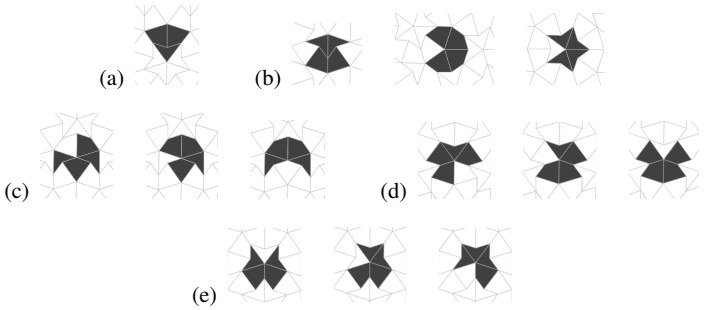


**Fig. 18.18** All the kite and dart block still lifes, kd-p1-3 and kd-p1-4, classified according to the valid vertex configurations: (a) the three cell ace *small block*; (b) the deuce *block*; the sun *block*; the star *block*; (c) the three jack *block*s; (d) the three queen *block*s; (e) the three king *block*s

#### 18.3.1.2 Kite and Dart Blocks

We can discover the isomorphic forms of blocks by exhaustive examination of the seven valid kite and dart vertices (Fig. 18.4).

Of these seven valid vertex configurations, one (the ace) has three cells meeting at the vertex. These three cells comprise a *small block* still life, Fig. 18.18a. There are no three cell still lifes in regular Life.

The deuce has four cells meeting at the vertex. These four cells comprise a block still life, Fig. 18.18bi.

All the other valid vertices have five cells meeting at the vertex; any four of these chosen to be "live" form a still life. Two of these vertices, the sun and the star, are symmetric, so have one block form, Fig. 18.18bii and biii. The others (the jack, queen, and king) each have three block variants (up to a reflection), Fig. 18.18c, d, and e.

#### 18.3.1.3 Rhomb Blocks

We can discover the isomorphic forms of blocks by exhaustive examination of the valid rhomb vertices (Fig. 18.5).

Of the eight valid vertex configurations, two (the Q and the D) have three cells meeting at the vertex. These three cells comprise a *small block* still life, Fig. 18.19a.

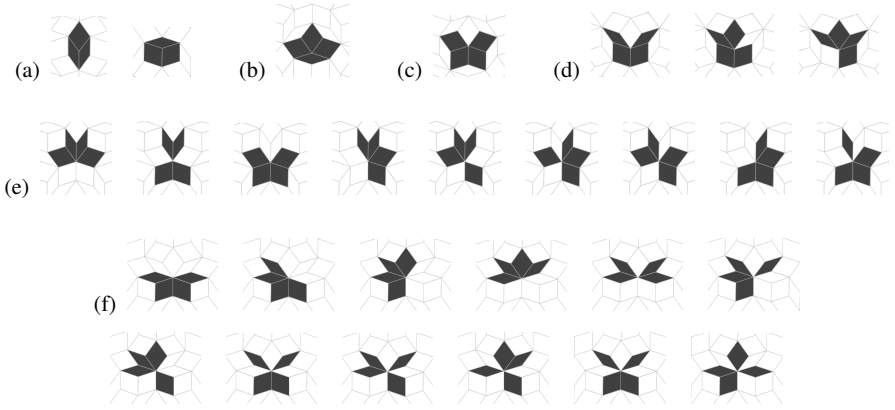


**Fig. 18.19** The rhomb block still lifes, r-p1-3 and r-p1-4: (a) the three cell Q *small block* and D *small block*; (b) the K *block*; (c) the S *block*, equivalent to the S5 *block*; (d) the three J *block*s; (e) the nine S4 *block*s; (f) a selection of the several S3 *block*s

One valid vertex (the K) has four cells meeting at the vertex. These four cells comprise a block, Fig. 18.19b.

Three valid vertices (the S5, the S, and the J) have five cells meeting at the vertex; any four of these chosen to be "live" form a still life. Two of these, the S5 and the S, are symmetric, so have one block form each (or one between them if vertex orientation is ignored), Fig. 18.19c. The J has three block variants (up to a reflection), Fig. 18.19d.

One valid vertex (the S4) has six cells meeting at the vertex; again, any four of these chosen to be "live" form a still life, Fig. 18.19e.

One valid vertex (the S3) has seven cells meeting at the vertex. There are many ways of choosing four of these to be "live" to form a still life; a selection is shown in Fig. 18.19f.

### *18.3.2 Five and More Cell Still Lifes*

There is a single regular Life five cell still life [9], the *boat* (Fig. 18.20). There is no Penrose life isomorphic to the boat: it is not possible to add a fifth stable cell to any of the tubs without giving an adjacent dead cell three live neighbours. All the identified Penrose five cell still lifes form chains (Fig. 18.21). (We call rings of cells, where every node has precisely two neighbours, *chains*.)

There are five regular Life six cell still lifes [9], the *snake*, the *ship*, the *barge*, the *beehive*, and the *carrier* (Fig. 18.22). The identified Penrose six cell still lifes form snakes (Fig. 18.23), chains (Fig. 18.24), or are disconnected (Fig. 18.25).

**Fig. 18.20** The regular Life five cell boat still life, l-p1-5

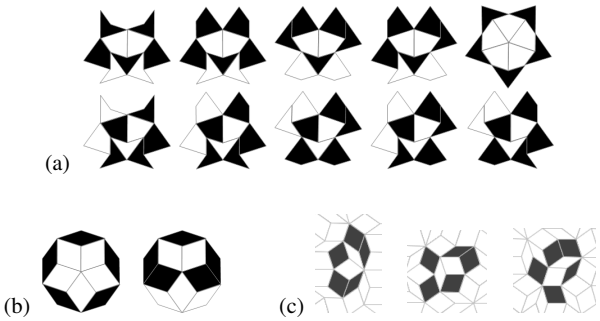


**Fig. 18.21** Some five cell still lifes, p1-5: a) kite and dart, from examining the ace and sun vertices; b) rhomb, from examining the S vertex; c) rhomb, found in the ash. We call the left pattern in b) an S-chain

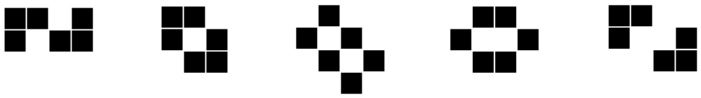

**Fig. 18.22** All the regular Life six cell still lifes, l-p1-6: the *snake*, the *ship*, the *barge*, the *beehive*, and the *carrier*

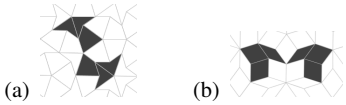


**Fig. 18.23** Some six cell Penrose snakes, p1-6: a) kite and dart; b) rhomb

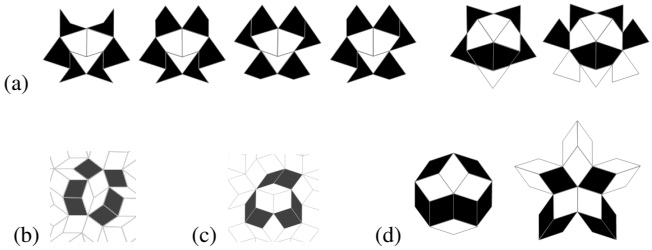


**Fig. 18.24** Some six cell Penrose still life chains, p1-6: a) kite and dart, from examining the ace and sun vertices; b) rhomb, found in the ash: note that it forms a chain around a Q vertex; c) rhomb, constructed by analogy to b, based on a D vertex; d) rhomb, from examining the S and S5 vertices

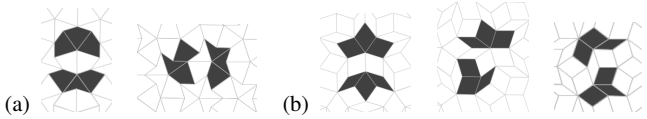


**Fig. 18.25** Some six cell disconnected still lifes, p1-6: a) kite and dart; b) rhomb

**Fig. 18.26** All the regular Life seven cell still lifes, l-p1-7: the *long snake*, the *fishhook*, the *long boat*, and the *loaf*

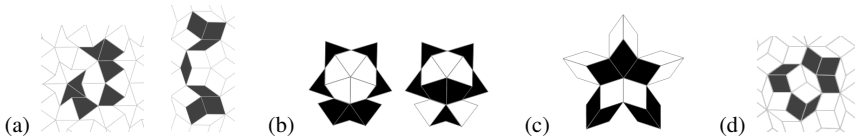


**Fig. 18.27** Some seven cell still lifes, p1-7: a) kite and dart 7-*snake* [7], and a rhomb 7-*snake*; b) kite and dart rings, from examining the sun vertex; c) rhomb ring, from examining the S5 vertex; d) rhomb ring

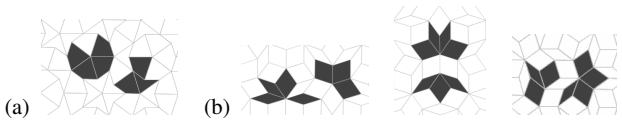


**Fig. 18.28** Some seven cell disconnected still lifes, p1-7: a) kite and dart; b) rhomb

There are four regular Life seven cell still lifes [9], the *long snake*, the *fishhook*, the *long boat*, and the *loaf* (Fig. 18.26). The identified Penrose seven cell still lifes form snakes or rings (Fig. 18.27) or are disconnected (Fig. 18.28).

There are nine regular Life eight cell still lifes [9], the *shillelagh*, the *hook with tail*, the *very long snake*, the *canoe*, the *long ship*, the *long barge*, the *pond*, the *tub with tail*, and the *cigar* (Fig. 18.29). The identified Penrose eight cell still lifes form chains, rings, and snakes (Fig. 18.30).

The number of regular still lifes increases sharply with size (from 10 with nine cells, 25 with 10 cells, 46 with 11 cells, on up to 112,243 with 20 cells [9]). We have no reason to believe that the Penrose still lifes do not similarly increase in numbers, however our ash searches have not revealed these. Our hand constructions are based on patterns already seen: such still life rings and chains are shown in Figs. 18.31–18.33. A systematic cataloguing search would need to consider all vertices, and all valid extensions of those vertices; the large number of block still lifes alone (Figs. 18.18 and 18.19) indicates this would be a significantly larger job than for regular Life.

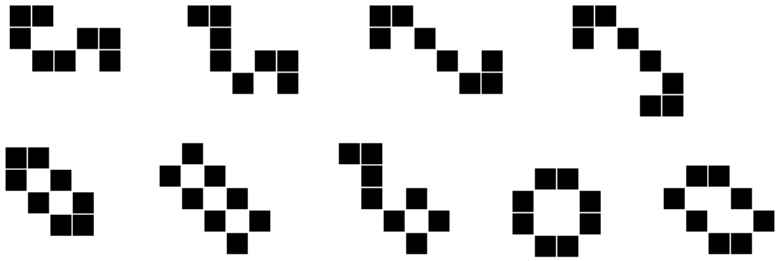

**Fig. 18.29** All the regular Life eight cell still lifes, l-p1-8: the *shillelagh*, the *hook with tail*, the *very long snake*, the *canoe*, the *long ship*, the *long barge*, the *tub with tail*, the *pond*, and the *cigar*

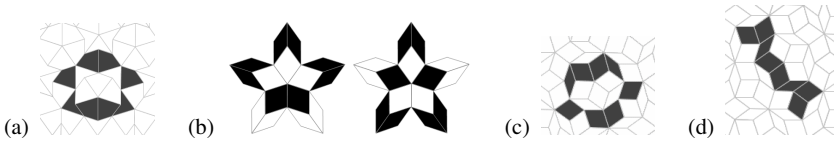


**Fig. 18.30** Some eight cell still lifes, p1-8: a) kite and dart chain; b) rhomb chain, and ring; c) another rhomb chain; d) rhomb *8-snake*

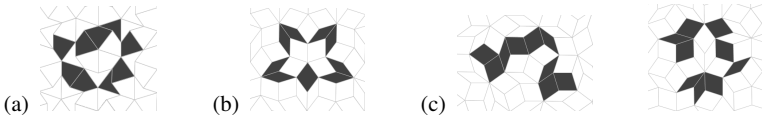


**Fig. 18.31** Some nine cell still lifes, p1-9: a) kite and dart chain chain; b) rhomb chain; c) rhomb *9-snake*s

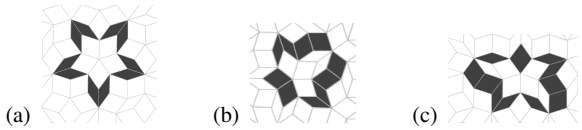


**Fig. 18.32** Some further still lifes: c) r-p1-10 chain, the *S5-chain*; d) another r-p1-10 chain; e) r-p1-11 chain

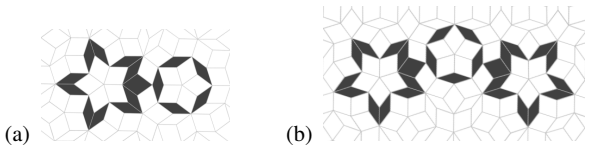


**Fig. 18.33** Some further still lifes: a) r-p1-16 ring pair from an *S*5-*S* combination; b) r-p1-27 ring triple from an *S*5-*S*-*S*5 combination

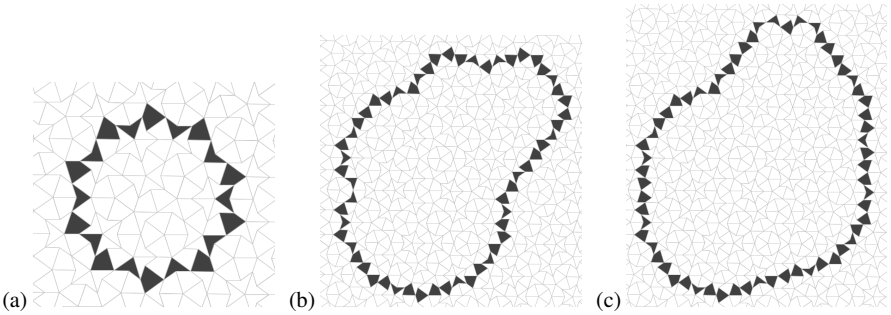


**Fig. 18.34** Some large kite and dart still life chains: (a) kd-p1-20; (b) kd-p1-57; (c) kd-p1-61

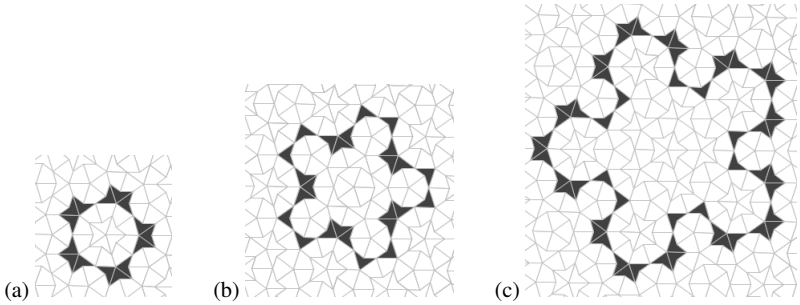


**Fig. 18.35** Dart still life rings: (a) kd-p1-15; (b) kd-p1-25; (c) kd-p1-55

**Fig. 18.36** The kd-p1-15 ring variant forms, from the three different extensions of the "Complete Star" $S_C$ (Fig. 18.15).

## *18.3.3 Large Rings*

### 18.3.3.1 Dart Rings

Large rings are not possible in regular Life, as a ring requires a "corner", which results in a dead site with three live neighbours. It was noted in [7] that large ring-shaped kite and dart still lifes can be formed. Some examples are shown in Fig. 18.34.

Arbitrarily large dart rings can be constructed, in the following way. Pick a dart that is not part of a Star $S_2$ vertex. Complete the "string" of darts that is formed from the darts in its generalised Moore neighbourhood (this is always possible, see Fig. 18.10). This string is a still life ring. Fig. 18.35 show example constructions.

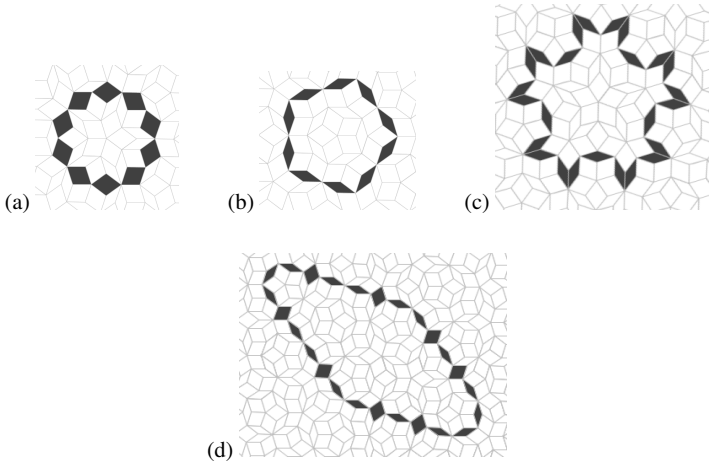


**Fig. 18.37** Some large rhomb still life chains: (a) r-p1-10, all thick rhombs; (b) r-p1-10, all thin rhombs; (c) r-p1-25, all thin rhombs; (d) r-p1-25, thick and thin rhombs. Note that our later classification in terms of oscillator graphs identifies the chains in (c) and (d) as isomorphic still lifes

Note that the dart ring in Fig. 18.35a occurs on the symmetric extension of the Complete Star $S_C$ (Fig. 18.15). There are variants of this ring, on the other two extensions, shown in Fig. 18.36.

#### 18.3.3.2 Rhomb Rings

Large still life chains can be formed in the rhomb tiling; some examples are shown in Fig. 18.37.

Arbitrarily large chains can be constructed, in the following way. Pick a thick rhomb that is not part of an *S* or *S*5 vertex. Complete the "ribbon" of thick rhombs that is formed from the two thick rhombs adjacent to its edges (all thick rhombs have precisely two such thick rhomb neighbours, see Fig. 18.9). Fig. 18.38 shows several such ribbons of thick rhombs.

Each thick rhomb ribbon is edged on the inside and outside by a thin rhomb ribbon; choose one of these. Delete any S-chains (Fig. 18.21bi). What remains is a thin rhomb still life chain. Figures 18.39 and 18.40 show two examples of such constructions, plus some variant forms.

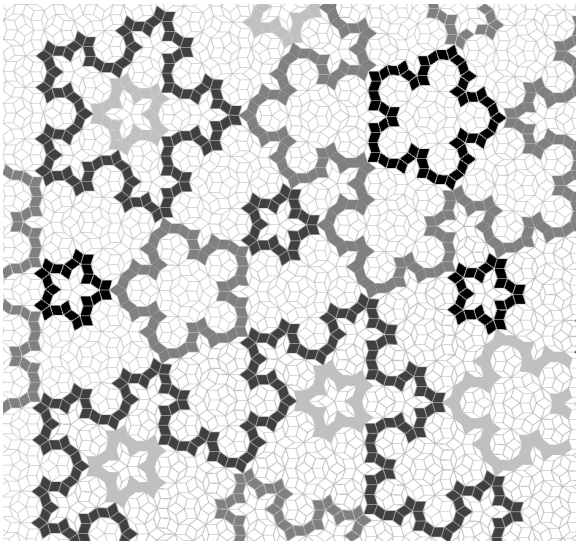

**Fig. 18.38** Some "ribbons" of thick rhombs

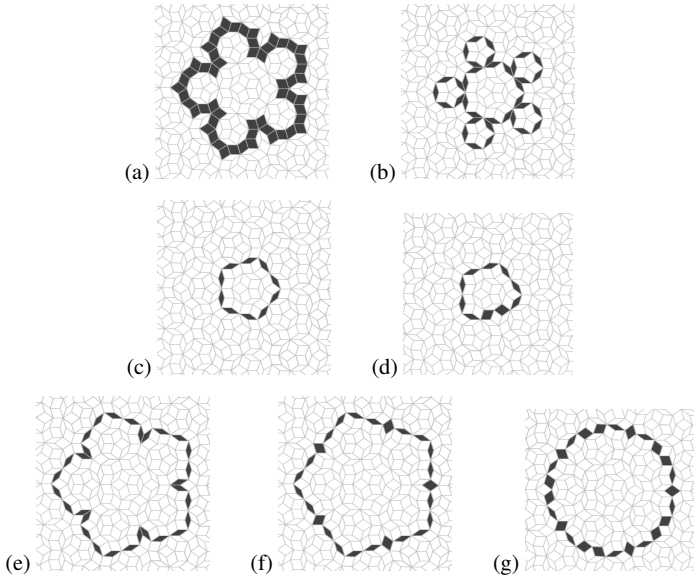


**Fig. 18.39** Constructing still life chains. (a) a thick rhomb ribbon; (b) the internal thin rhomb ribbon. (c) the internal ribbon with the *S*-chains removed: r-p1-10, a still life chain; (d) a variant r-p1-10, constructed by "folding in" a pair of thin rhombs to become thick rhombs; (e) the external thin rhomb ribbon: r-p1-30, a still life chain; (f) a modification r-p1-25, constructed by replacing folding the thin rhomb "elbow" pair into a single thick rhomb, five times (this has similarities to the relationship between the regular Life *pond*, Fig. 18.29, and *loaf*, Fig. 18.26, with two squares folded into one); (g) a variant r-p1-25, constructed by "folding in" five pairs of thin rhombs to pairs of thick rhombs. A modification of this "folding" construction can be applied to convert chains into $p2$ oscillators (see Fig. 18.61)

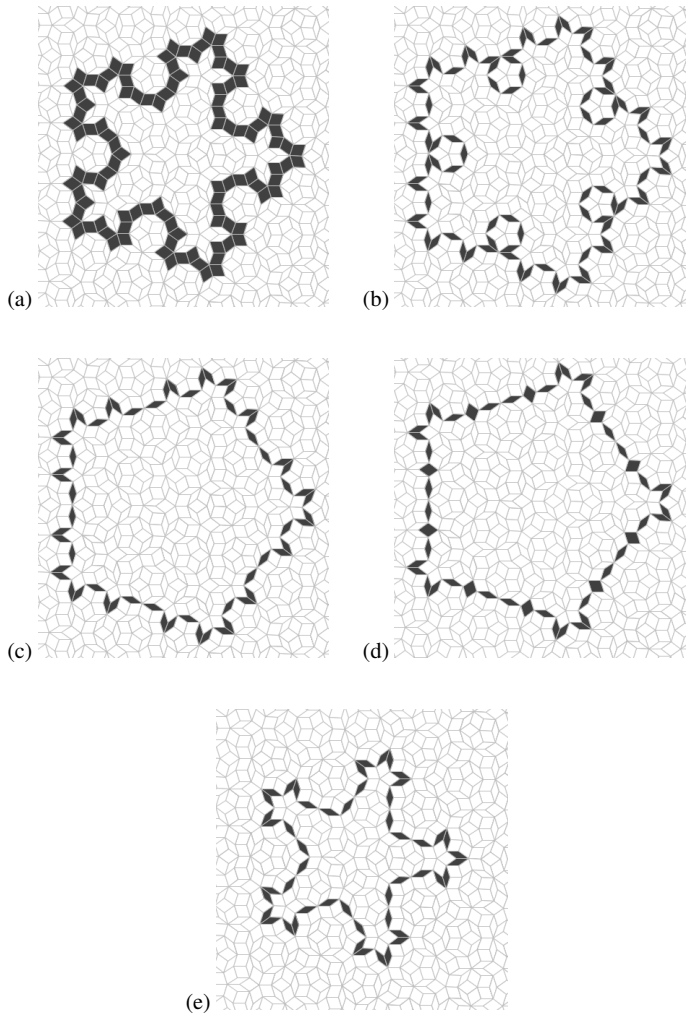


**Fig. 18.40** Constructing still life chains. Top row: (a) a thick rhomb ribbon; (b) the external thin rhomb ribbon. (c) the external ribbon with the *S*-chains removed: r-p1-60, a still life chain; (d) r-p1-50, a modification of c; (e) the internal thin rhomb ribbon: r-p1-60, a still life chain

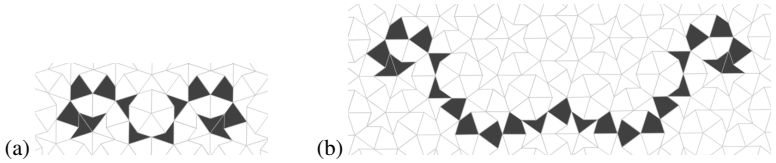


**Fig. 18.41** kite and dart still life snakes: a) kd-p1-14: a *14-snake*; b) kd-p1-25: a *25-snake*

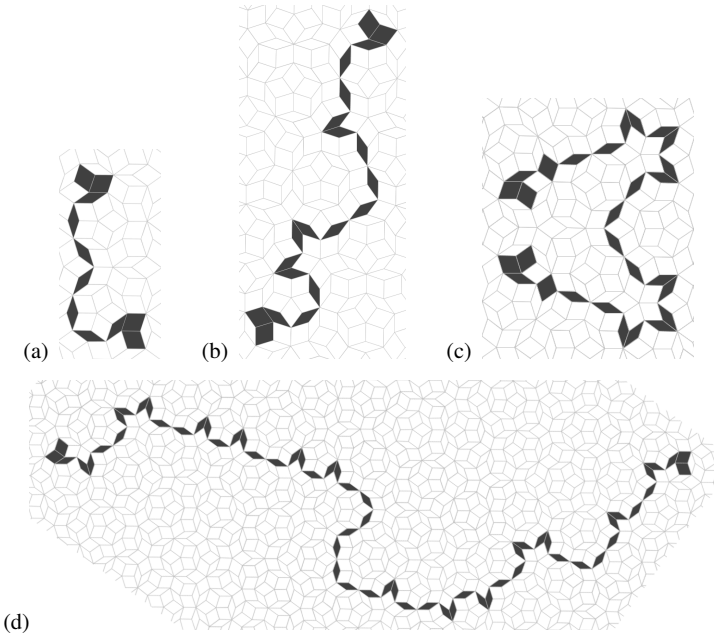


**Fig. 18.42** rhomb still life snakes, constructed from a combination of thin rhomb chains and K vertex terminators: a) r-p1-11: an *11-snake*; b) r-p1-20: a *20-snake*; c) r-p1-28: a *28-snake*; d) r-p1-61: a *61-snake*

### *18.3.4 Large Snakes*

Long linear kite and dart still lifes can be formed; some examples are shown in Fig. 18.41. By combining the rhomb chain construction process with the linear still life termination on a K vertex (Fig. 18.5), long linear rhomb still lifes can also be constructed, see Fig. 18.42.

There is potential for using long snakes when making larger "machines". Disrupting a snake at some site, such as one end, causes it to "disintegrate" from that site, propagating the disruption along the snake, somewhat like a (messy) "fuse". Or if some activity were to break a long chain, leaving a terminator on one end, then a circular fuse would burn, with activity returning to the original point some time later. By choosing the length of the chain, this could provide a form of timer.

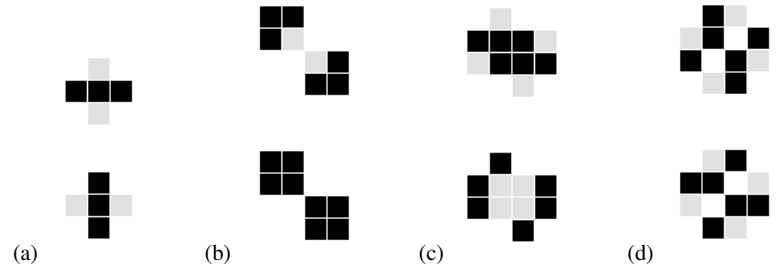


**Fig. 18.43** The smallest regular life $p2$ oscillators: (a) the *blinker*, l-p2-5-3; (b) the *beacon*, l-p2-8-6; (c) the *toad*, l-p2-10-6; (d) the *clock*, l-p2-10-6

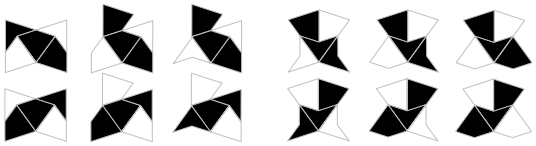

**Fig. 18.44** The six distinct kite and dart plinkers: kd-p2-5-3.

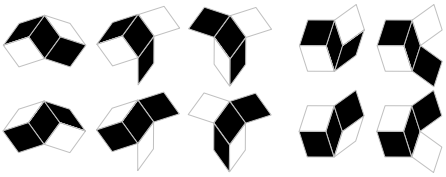

**Fig. 18.45** The five distinct rhomb plinkers: r-p2-5-3

## 18.4 Period 2 Oscillators

### *18.4.1 Blinkers and Plinkers*

A regular Life "blinker" is a period 2 oscillator comprising a line of three live cells. If in generation 0 it is a horizontal line, then in generation 1 it is a vertical line sharing the same central cell (Fig. 18.43a).

There are isomorphic three cell, $p2$ *plinkers*[3] in Penrose life. A plinker can exist at any cell in the tiling[4].

[3] These particular oscillators were dubbed "plinkers" in [7], and we continue that usage here, as an exception to our naming convention.

[4] *Proof*: Consider any cell, which will be the 'central' cell of the plinker. Consider a pair of opposite (non-adjacent) edges of this central cell. Consider the two cells adjacent to this pair of edges. Make these two cells and the central cell alive, and all other cells dead. The result is a plinker, oscillating between the chosen pair of cells, and the pair of cells adjacent to the other two edges of the central cell.

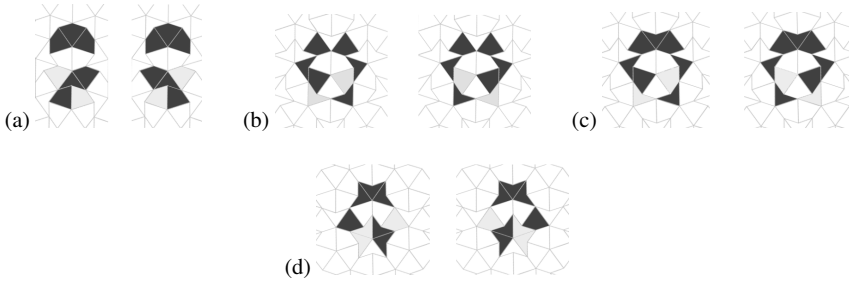


**Fig. 18.46** Further $p2$ kite and dart oscillators: (a) kd-p2-8-6, a disconnected oscillator, constructed from the still life of Fig. 18.25ai by converting the lower 3-block to a plinker; (b) kd-p2-8-6, the *marcher*; (c) kd-p2-9-7, the *crowned marcher*; (d) kd-p2-9-6

**Fig. 18.47** kd-p2-10-6: the *hollow clock*, two isomorphic $p2$ kite and dart oscillators related to, but not the same as, the regular Life $p2$ oscillator the *clock* (Fig. 18.43d)

**Fig. 18.48** kd-p2-12-9: the *fast shuffler*, which exists in a symmetric and asymmetric variant form, from the different extensions of the "Complete Star" $S_C$ (Fig. 18.15)

The set of generalised von Neumann neighbourhoods (Figs. 18.8 and 18.9) can be used to enumerate the complete set of distinct plinkers. There are six plinkers for kites and darts (Fig. 18.44, as noted in [7]), and five for rhombs (Fig. 18.45). Each plinker has 3 cells alive at any time, and a total of 5 distinct cells used.

### *18.4.2 Other p2 Oscillators*

All six varieties of the kite and dart *plinker* (Fig. 18.44) were discovered while exploring the behaviour of the rules in [7]. In our subsequent searches new larger $p2$ kite and dart oscillators (Fig. 18.46–18.49) have been discovered.

The rhomb tiling also exhibits its own zoo of oscillators. In addition to the rhomb plinkers (Fig. 18.45), there are further $p2$ rhomb oscillators (Figs. 18.50–18.60). Large $p2$ rhomb oscillators can be constructed from rhomb chain still lifes (Figs. 18.61–18.62).

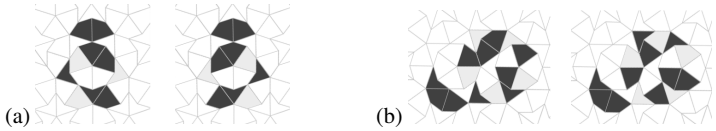


**Fig. 18.49** Further $p2$ kite and dart oscillators: (a) kd-p2-10-7; (b) kd-p2-15-10

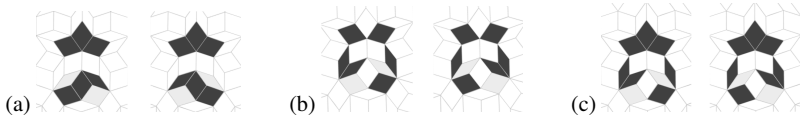


**Fig. 18.50** $p2$ rhomb oscillators: (a) r-p2-8-6, a disconnected oscillator, constructed from the still life of Fig. 18.25bi by converting the lower 3-block to a plinker (the same construction can be applied to the still life of Fig. 18.28bi, but the resulting 4-block and plinker are each individually stable, so the result does not count as a disconnected oscillator); (b) r-p2-8-6, the *marcher*; (c) r-p2-9-7, the *crowned marcher*

**Fig. 18.51** r-p2-10-6, two isomorphic forms of the *hollow clock*

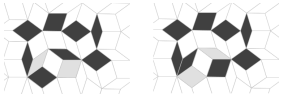

**Fig. 18.52** r-p2-10-8


**Fig. 18.53** (a) r-p2-11-8; (b) r-p2-11-9.

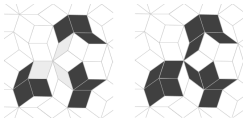

**Fig. 18.54** r-p2-12-9: the *big beacon*, named by analogy to the regular Life $p2$ *beacon* (Fig. 18.43b), but here with three components instead of two

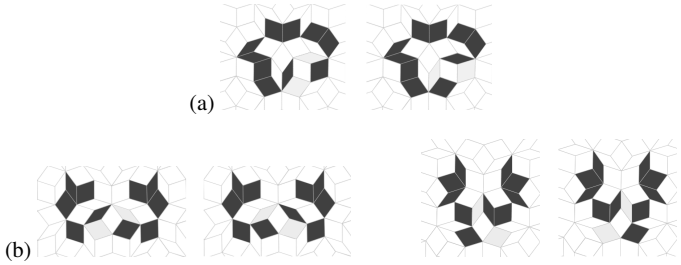


**Fig. 18.55** r-p2-12-10: (a) a ring oscillator; (b) two isomorphic oscillators. The way that (a) is shown to be different from (b), and that the two forms of (b) are found to be isomorphic, is described later

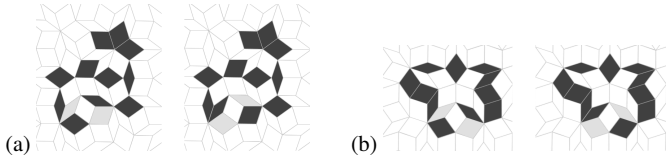


**Fig. 18.56** r-p2-13-11: (a) Note that this is the r-p2-10-8 with three extra cells added; (b) note the relationship to r-p1-11 of Fig. 18.31d.

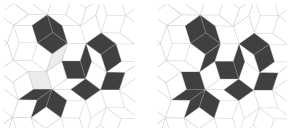

**Fig. 18.57** r-p2-14-12.

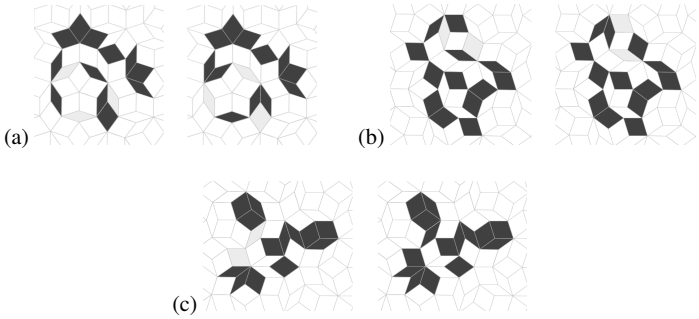


**Fig. 18.58** (a) r-p2-15-12; (b) r-p2-15-13; (c) r-p2-15-13.

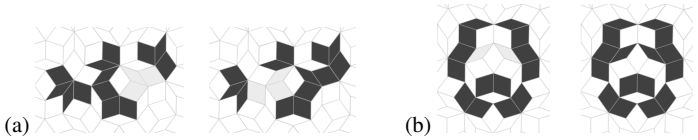


**Fig. 18.59** (a) r-p2-16-13; (b) r-p2-16-14.

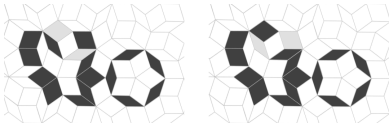

**Fig. 18.60** r-p2-17-14.

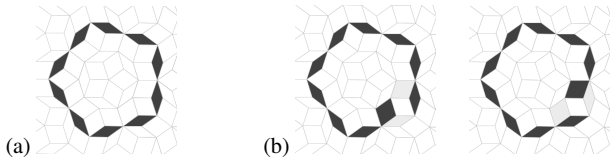


**Fig. 18.61** Starting from (a) a still life rhomb chain r-p1-10, we can construct (b) a corresponding $p2$ rhomb ring oscillator r-p2-12-10 (isomorphic to Fig. 18.55a). This construction can be applied to all the large rhomb chains of Fig. 18.39.

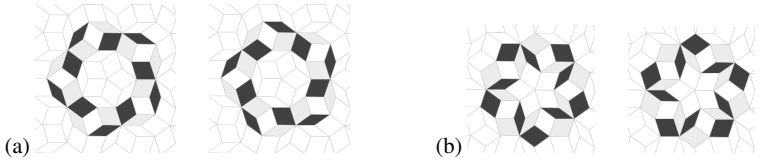


**Fig. 18.62** The construction of Fig. 18.61 can be applied to more than one site in a chain: (a) a maximal $p2$ construction applied to Fig. 18.61a; (b) a maximal construction applied to Fig. 18.37a. These are both r-p2-20-10; many variants exist.

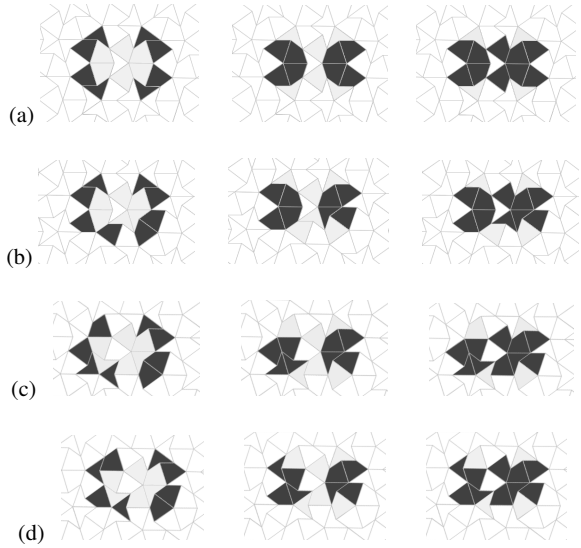


**Fig. 18.63** kd-p3-14-8: a) the kite and dart *breather*; b) an isomorphic variant breather; c) a further isomorphic variant breather; d) a fourth isomorphic variant breather.

**Fig. 18.64** kd-p3-15-10: the *wagger*.

## 18.5 Period 3 Oscillators

Four essentially different $p3$ kite and dart oscillators have been discovered: the *breather* with four isomorphic forms (Fig. 18.63), the *wagger* (Fig. 18.64), a disconnected oscillator with two isomorphic forms (Fig. 18.65), and a more irregular oscillator (Fig. 18.66).

Fewer period three rhomb oscillators have been found. There is a rhomb oscillator isomorphic to the kite and dart $p3$ *breather* (Fig. 18.67), and a further $p3$ oscillator (Fig. 18.68).

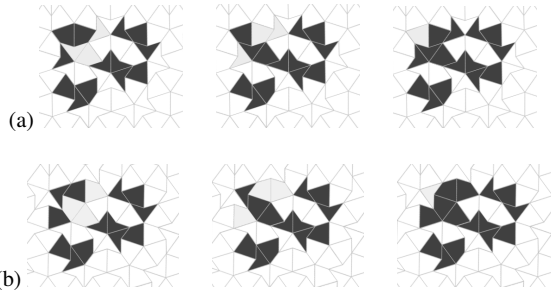


**Fig. 18.65** kd-p3-16-13: a) a disconnected oscillator; b) a variant form.

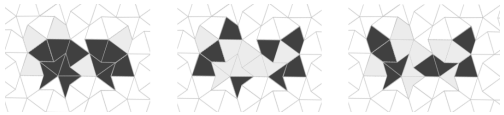

**Fig. 18.66** kd-p3-17-8.

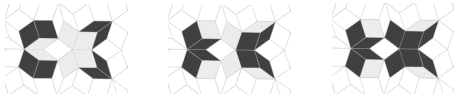

**Fig. 18.67** r-p3-14-8: the rhomb *breather*.

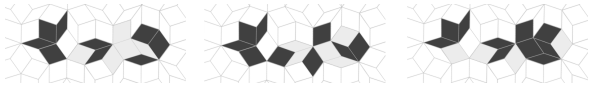

**Fig. 18.68** r-p3-14-9.

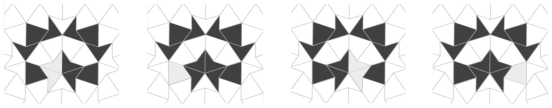

**Fig. 18.69** kd-p4-12-10: the *shuffler*. Note the underlying period 2 behaviour, combined with a reflection.

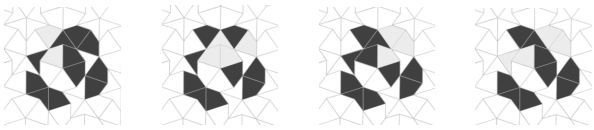

**Fig. 18.70** kd-p4-13-9

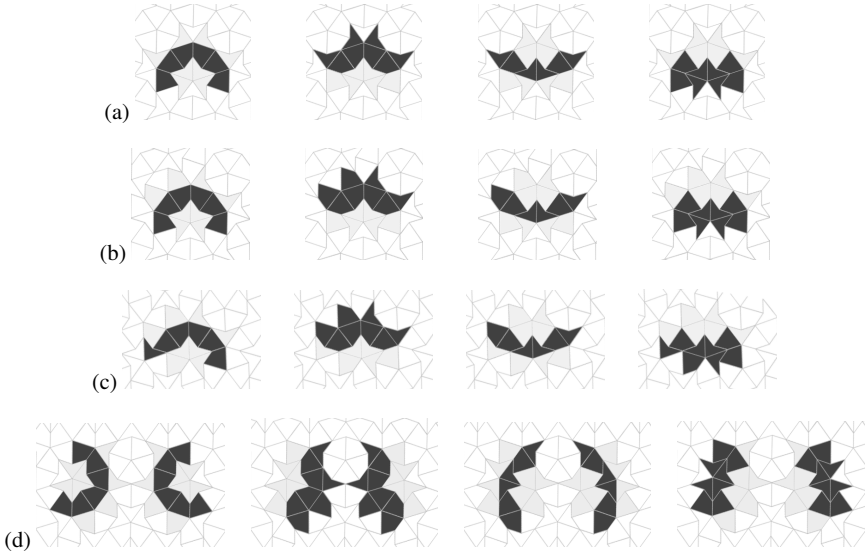


**Fig. 18.71** kd-p4-14-6: a) the *bat*, from [7]; b) the variant *asymmetric bat*, from a different extension of the "Complete Star" $S_C$ (Fig. 18.15); c) an isomorphic oscillator, not located on a "Complete Star" $S_C$; d) the *bat-to-bat*, where the left and right halves are each an asymmetric bat oscillator, but the bats touch at step two, kd-p4-28-12.

## 18.6 Period 4 Oscillators

Several kite and dart $p4$ oscillators are shown in Figs. 18.69–18.72.

The 6 cell kite and dart $p4$ *bat* (Fig. 18.71a) was discovered in [7]. Here we also see variants: a $p4$ *asymmetric bat*, a variant form located on a different Complete Star extension (Fig. 18.71b), and a further isomorphic *bat*, not located on a Complete Star vertex extension (Fig. 18.71c). There is also a *bat-to-bat* oscillator (Fig. 18.71d), which is two bat oscillators that actually touch at one point.

Period 4 rhomb oscillators are shown in Figs. 18.73–18.77.

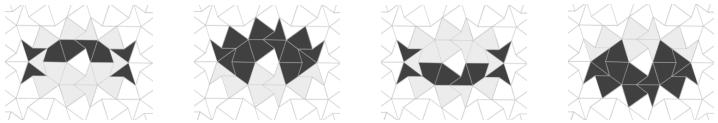

**Fig. 18.72** kd-p4-22-8: the *hedgehog*. Note the underlying period 2 behaviour, combined with a reflection.

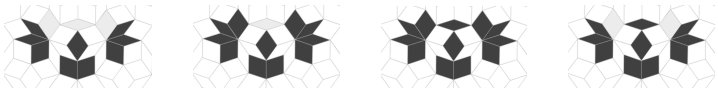

**Fig. 18.73** r-p4-12-9: the *clown*.

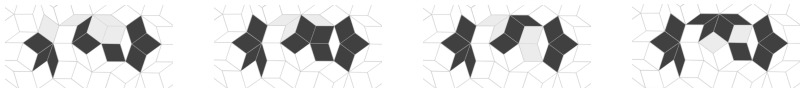

**Fig. 18.74** r-p4-14-9.

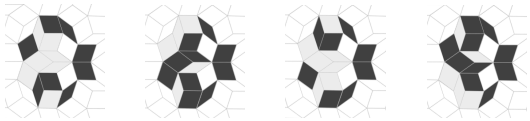

**Fig. 18.75** r-p4-15-8: the *goldfish*. Note the underlying period 2 behaviour, combined with a reflection.

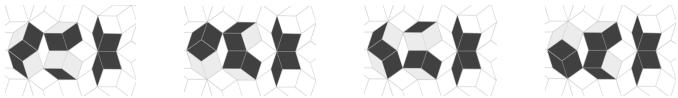

**Fig. 18.76** r-p4-15-10. Note the underlying period 2 behaviour, combined with a reflection.

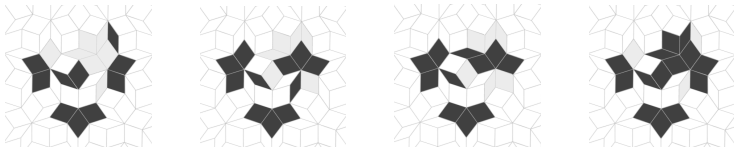

**Fig. 18.77** r-p4-17-10: the *pirate*.

**Fig. 18.78** kd-p5-15-8: the *ninja*. Note the constant pattern undergoing a $4\pi/5$ rotation per timestep.

**Fig. 18.79** kd-p5-16-10, a disconnected oscillator: the *drummer*.

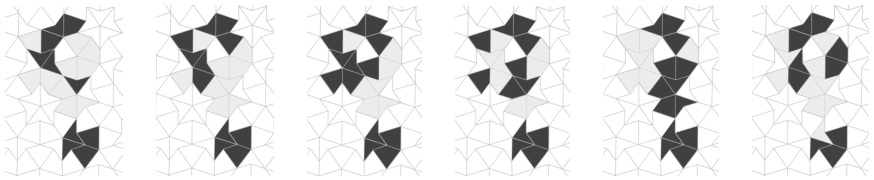

**Fig. 18.80** kd-p6-20-10: the *tickler*.

## 18.7 Higher Period Oscillators

### *18.7.1 Kite and Dart High Period Oscillators*

One of the interesting discoveries in [7] was the existence of spatially small, but relatively long period, oscillators in the kite and dart tiling, relative to the regular lattice. An $p8$ oscillator (Fig. 18.81), and the $p15$ *dancer* (Fig. 18.84), were discovered while exploring the behaviour of the rules.

In subsequent searches new kite and dart long period oscillators have been discovered: the $p5$ *ninja* (Fig. 18.78) and *drummer* (Fig. 18.79), the $p6$ *tickler* (Fig. 18.80, the $p9$ *moustaches* (Fig. 18.82), and the $p11$ *malformed bat* (Fig. 18.83). Many of these exist in multiple isomorphic forms; we no longer display variant forms, for reasons of space, but show only the most "symmetric" variant found.

Note that when a period is composite, the oscillator may exhibit subperiods (for example, the $p4$ *shuffler* has a $p2$ behaviour that is then reflected; the $p15$ dancer has a $p3$ behaviour that undergoes a five-fold rotation), or not (the $p6$ *tickler* and the $p9$ *moustaches* have no obvious sub-periodic behaviours).

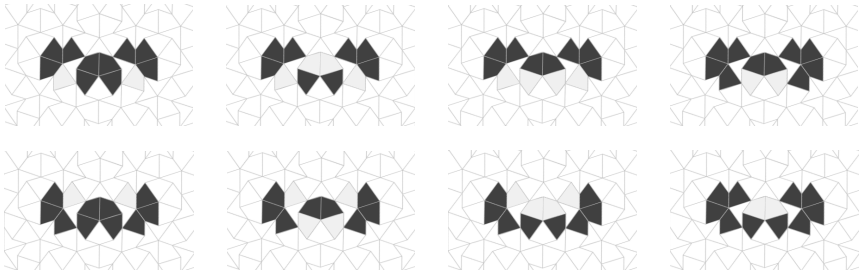

**Fig. 18.81** kd-p8-12-8, from [7].

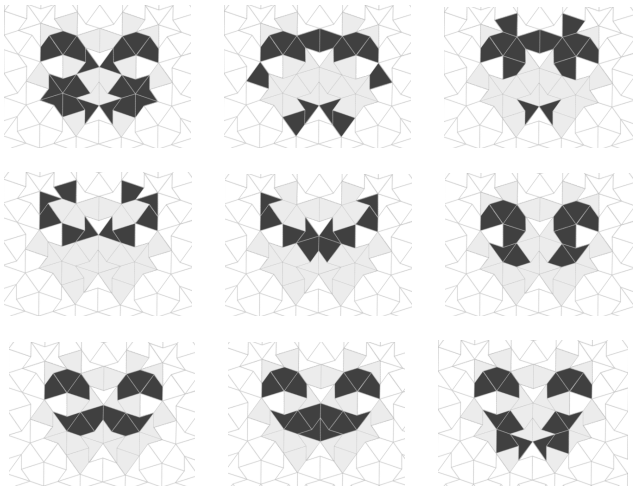

**Fig. 18.82** kd-p9-36-10: the *moustaches*.

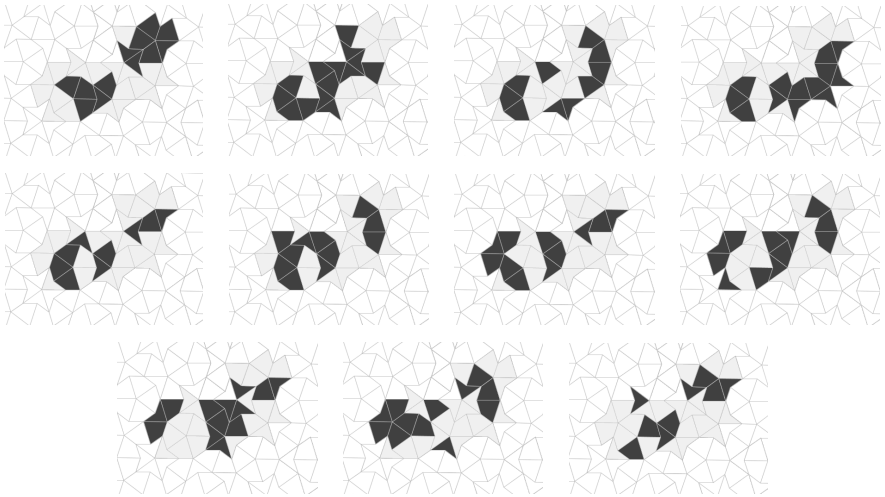

**Fig. 18.83** kd-p11-31-9: the *malformed bat*. It has similarities to the *p*4 *bat* [7].

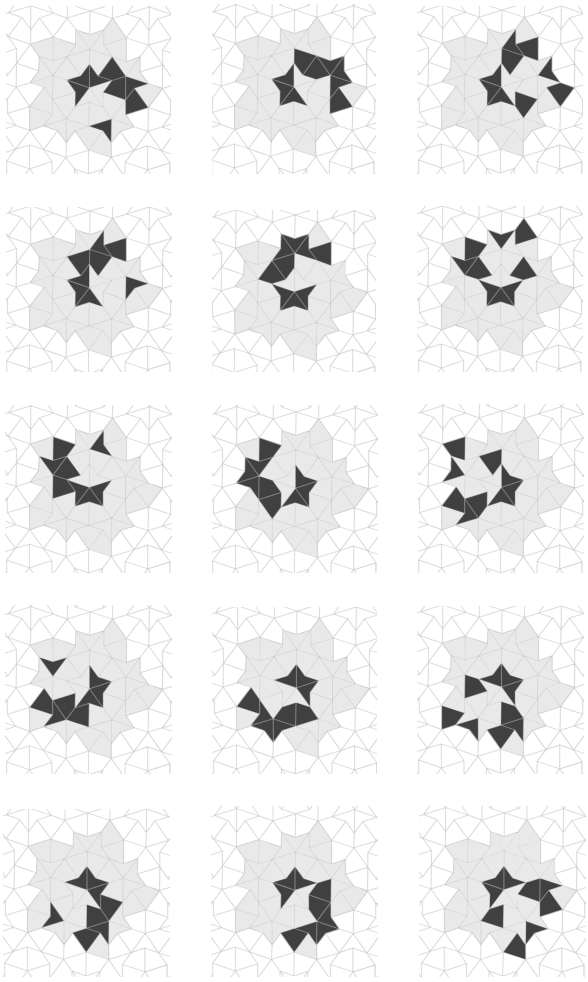

**Fig. 18.84** kd-p15-40-8: the *dancer*, from [7]. Note the underlying period 3 behaviour, combined with a 5-fold rotation. The dancer exists in three variant forms (not shown), from the three different extensions of the "Complete Star" $S_C$ (Fig. 18.15).

**Fig. 18.85** r-p5-14-7, a disconnected $p5$ rhomb oscillator: the *juggler*.

**Fig. 18.86** r-p5-15-8: the *ninja*.

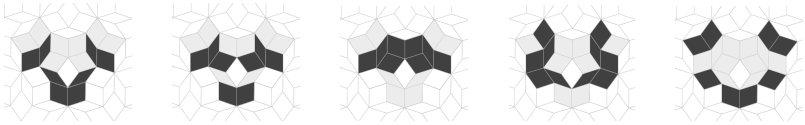

**Fig. 18.87** r-p5-16-8: the *jumper*

**Fig. 18.88** r-p5-18-11, another disconnected $p5$ rhomb oscillator.

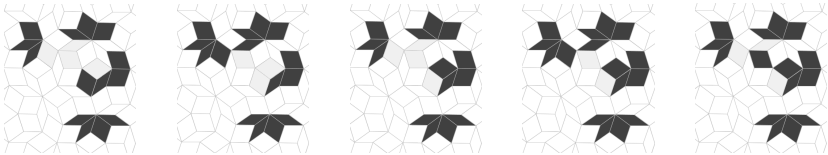

**Fig. 18.89** r-p5-18-14, a further disconnected $p5$ rhomb oscillator.

### *18.7.2 Rhomb High Period Oscillators*

Higher period rhomb oscillators are shown in Figs. 18.85–18.96. One of these, the *ninja* (Fig. 18.86) is isomorphic to the kite and dart ninja (Fig. 18.78).

These high period complex rhomb oscillators help demonstrate that the rhomb-based CA, whilst having statistically significantly different statistical behaviour from the kite and dart-based CA under Game of Life rules [11], also exhibits complex and interesting behaviour.

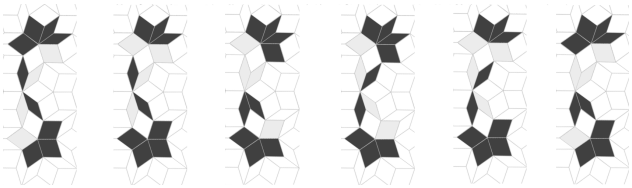

**Fig. 18.90** r-p6-13-9

**Fig. 18.91** r-p7-19-7: the *hattipper*

**Fig. 18.92** r-p7-23-8, related to the *hattipper*.

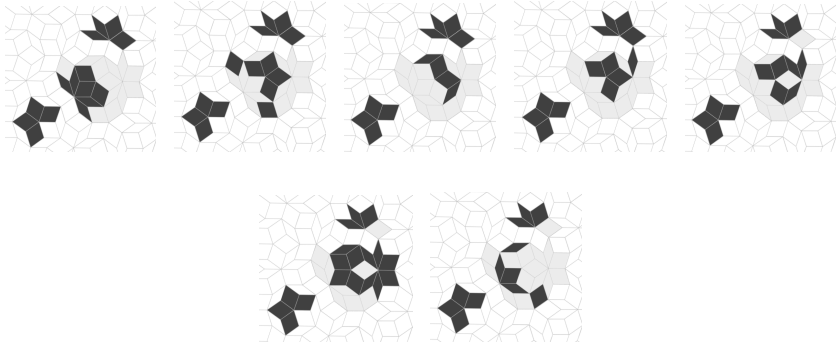

**Fig. 18.93** A disconnected relation of the *hattipper*.

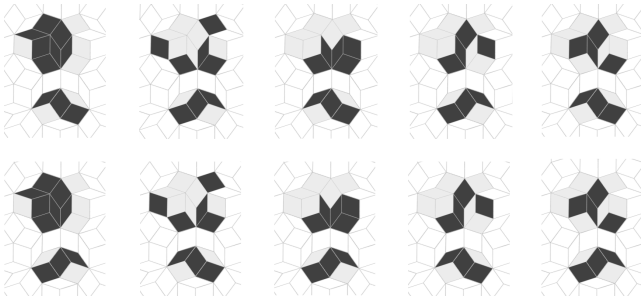

**Fig. 18.94** A disconnected $p10$ rhomb oscillator, the *long juggler*. Note the combination of a period 5 behaviour (upper cells of the oscillator, identical to the upper portion of the $p5$ juggler) and a period 2 plinker (lower cells). Compare Figs. 18.85 (which shows the underlying $p5$ *juggler* and 18.50a (which shows a similar construction of a $p2n$ oscillator from an underlying $pn$ oscillator).

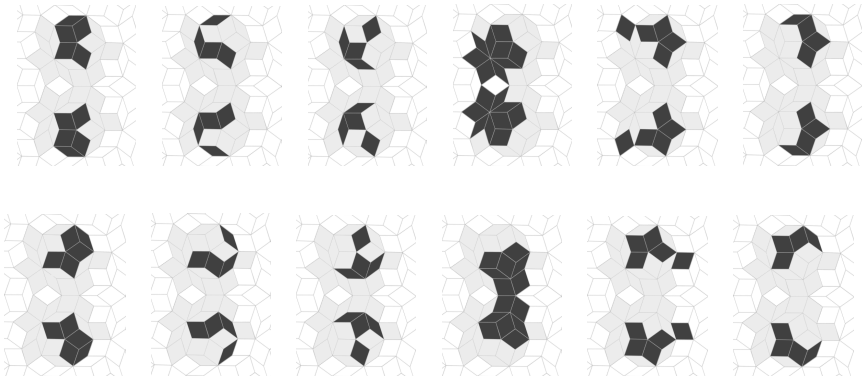

**Fig. 18.95** r-p12-36-8: the *reflected bouncer*.

**Fig. 18.96** r-p12-33-6: *fireworks*. Note the underlying period 6 behaviour, combined with a reflection.

## 18.8 Oscillator Analysis

Although some of the oscillators presented earlier were constructed manually, many (particularly the more irregular and long period oscillators) were discovered by random search, by examining the ash left from a set of CA runs starting from random initial conditions. The majority of such runs end with mostly small, low period oscillators (blocks, small rings, and plinkers, and, in the kite and dart case, the occasional bat or dancer; note that random search will be biassed against finding oscillators with few or no states leading to them). In order to analyse the large number of runs needed to find the rarer oscillators, it is necessary to have algorithms to detect the oscillators, and to identify and classify them. The irregular Penrose tiling makes this more difficult than for regular Life.

For identification, we have the code name, but this does not uniquely identify oscillators (although collisions are rare). We would like to classify to oscillators to identify those that are isomorphic (for example, the plinkers in Figs. 18.44 and 18.45, or the bats in Fig. 18.71), and those that are truly different (for example, the p1-4 tubs in Fig. 18.17 versus the p1-4 blocks in Figs. 18.18 and 18.19, or the different r-p2-12-10 oscillators in Fig. 18.55).

We have the general requirement that the algorithms should be efficient. The computational overheads of our lazy tiling algorithm to define the Penrose grid (described in [10]), and of running the CA rules, are reasonable; the more efficiently we can detect oscillators the better we can explore oscillator space.

Oscillator analysis requires three steps: (i) quiescence detection; (ii) oscillator detection; (iii) oscillator classification.

Quiescence detection is straightforward: we detect when the behaviour of the CA becomes periodic (this would not work if ever a propagating structure like a regular Life glider were to form, but this has never been the case so far). Periodic behaviour is detected by a comparison between each new state of the CA to every old state[5]. Detection and classification algorithms are described in the following sections.

### *18.8.1 Oscillator Detection*

Given a large patch of tiling containing many potential oscillators, we need a way to identify each individual oscillator, separate from the rest.

[5] We have investigated other methods of period detection. Application of Floyd's algorithm [8, p7] uses two versions of the CA: the first is updated normally, the second is updated twice for every update of the first; when they become equal the automaton has cycled. This method offers $O(1)$ space, and $O(n)$ time, where $n = t_q + p$ is the time to quiescence plus the period of the oscillation. Our method requires $O(n)$ space and $O(n^2)$ time (requiring a triangle number of comparisons, so $(n^2+n)/2$). However in practice our method is much faster as the size of the tiling greatly dominates the lifetimes involved here. The costs of running a second CA, even implemented just as a second state in each cell, is quite a performance hindrance.

For the purposes of detection, define an oscillator $O$ with period $p$ as a set of $n$ pairs, each pair being a cell and its sequence of $p$ states:

$$O = \{(c_0, S_p(c_0)), (c_1, S_p(c_1)), \ldots, (c_n, S_p(c_n))\} \tag{18.2}$$

where $S_p(c) = \sigma_0(c), \sigma_1(c), \ldots, \sigma_{p-1}(c)$ and $\sigma_t(c)$ is the state of cell $c$ at time $t$; $\sigma_t(c) = \blacksquare$ if the cell is alive at time $t$ and $\sigma_t(c) = \square$ if the cell is dead at time $t$. We say $c \in O$ to mean cell $c$ is one of the cells in the pairs contained in $O$. For all cells $c \in O$, $\sigma_0(c) = \sigma_p(c)$.

For example, consider a particular plinker $O_\pi$ (one of those in Fig. 18.44 or 18.45). It has five cells, the central one on all the time, and two pairs on and off in antiphase. Labelling the central cell $c_0$, one pair of cells $c_1$ and $c_2$, and the other pair $c_3$ and $c_4$, we have

$$O_\pi = \{(c_0, \blacksquare\blacksquare), (c_1, \blacksquare\square), (c_2, \blacksquare\square), (c_3, \square\blacksquare), (c_4, \square\blacksquare)\} \tag{18.3}$$

Define $N(c)$, the oscillator neighbourhood of cell $c$, to be the set of cells in the oscillator, $n \in O$, that are in the generalised Moore neighbourhood of $c$. For the plinker $O_\pi$, we have $N(c_0) = \{c_0, c_1, c_2, c_3, c_4\}$ and $N(c_1) = \{c_0, c_1, c_3, c_4\}$. Note that the neighbourhood relationship is symmetric: $c_1 \in N(c_2) \Leftrightarrow c_2 \in N(c_1)$.

Define the live and dead neighbourhoods, $N_t^{\blacksquare}(c)$ and $N_t^{\square}(c)$, by:

$$N_t^s(c) = \{n \in N(c) \mid \sigma_t(n) = s\} \tag{18.4}$$

where $s \in \{\square, \blacksquare\}$. So $N_t^{\square}(c)$ is the set of the oscillator's dead cells in the neighbourhood of $c$ at time $t$, and $N_t^{\blacksquare}(c)$ is the set of the oscillator's live cells. Note that $N(c) = N_t^{\square}(c) \cup N_t^{\blacksquare}(c)$, for any $t$.

For the plinker $O_\pi$, the live and dead neighbourhoods are:

$$N_0^{\square}(c_0) = \{c_3, c_4\} \; ; \; N_0^{\blacksquare}(c_0) = \{c_0, c_1, c_2\} \tag{18.5}$$

$$N_1^{\square}(c_0) = \{c_1, c_2\} \; ; \; N_1^{\blacksquare}(c_0) = \{c_0, c_3, c_4\} \tag{18.6}$$

$$N(c_0) = N_0^{\square}(c_0) \cup N_0^{\blacksquare}(c_0) = N_1^{\square}(c_0) \cup N_1^{\blacksquare}(c_0) = \{c_0, c_1, c_2, c_3, c_4\} \tag{18.7}$$

We take a minimalist view of an oscillator: a cell is included in an oscillator if its removal would destroy the oscillator. So when considering the cells of an oscillator alone, they may have incomplete neighbourhoods, but sufficient neighbourhoods to allow correct oscillation. When the oscillator is considered as part of a larger tiling there may be cells bordering the oscillator that must always be dead for the oscillator to exist; these always dead bordering cells are not considered part of the oscillator, since, as will become clear, they are not actively influencing the oscillator. This approach to oscillator detection provides a clean platform to perform the subsequent oscillator classification.

The oscillator detection is performed on the automaton ash in two stages: (A) an assignment of cells to potential oscillators; (B) a removal of non-necessary cells from the potential oscillators.

1: **for** $t = 0$ to $p-1$ {each automaton state in the ash} **do**
2: **while** there is a live cell $c$ that has not been processed yet **do**
3: **if** $c \notin \Omega$ {$c$ is not contained in any oscillator} **then**
4: {create a new oscillator $O$ containing just $c$ and its current state}
$O := \{(c, \sigma_t(c))\}; \Omega := \Omega \cup \{O\}$
5: **else** { $c$ is in an oscillator, with $(c,S) \in O$ }
6: $S := S + \sigma_t(c)$ {update $c$'s state list with $c$'s current state}
7: **end if**
8: **for** each $n \in N(c)$ {each of cell $c$'s neighbourhood cells} **do**
9: **if** $\sigma_t(n) =$ ■ {$n$ is alive}
or $\sigma_t(n) =$ □ and $|N_t^{\blacksquare}(n)| \geq 3$
{$n$ is dead and has three or more live cells in its neighbourhood} **then**
10: $O := O \cup \{(n, \sigma_0(n), \ldots, \sigma_t(n))\}$ {add $n$ to $O$}
11: **end if**
12: **end for**
13: **if** any of the cells $n$ added to $O$ are already a member of another oscillator $R$ **then**
14: $O := O \cup R; \Omega := \Omega - \{R\}$ {combine $O$ and $R$}
15: **end if**
16: continue recursively processing all neighbourhood cells $n$ added to $O$
17: **end while**
18: **end for**

**Fig. 18.97** Detection stage A algorithm

#### 18.8.1.1 Detection Stage A

For each CA ash state, a breadth-first search starting from each live cell is performed, to construct the set of all oscillators, $\Omega$, as described in the algorithm in Fig. 18.97.

Examination of the two conditions under which dead cells are added to an oscillator (the "or" condition of step 9 of Fig. 18.97) reveals why a second removal stage (B) is necessary in oscillator detection to remove certain dead cells.

If the dead cell $n$ has exactly three live neighbours, $|N_t^{\blacksquare}(n)| = 3$, then $n$ will become alive on the next automaton iteration, and so is added to $O$. It is added immediately, to guard against the following situation: all three neighbours that caused $n$'s birth die in the same iteration, no more neighbours of $n$ are born, and all now dead neighbours of $n$ fail to satisfy the conditions under which dead cells are added to an oscillator. So $n$ would not be associated with the correct oscillator if it were not for the $|N_t^{\blacksquare}(n)| = 3$ condition. This situation arises in the $p12$ fireworks (Fig. 18.96).

If the dead cell $n$ has more than three live neighbours, $|N_t^{\blacksquare}(n)| > 3$, then the cells of the oscillator are exerting an influence on the dead cell, forcing it to stay dead (from 'overcrowding'). Adding it to $O$ allows detection of oscillators with disjoint sections that interact via a forced dead cell. However, adding it may also add cells that are forced dead on the interior of the oscillator, or on the boundary but which may also connect coincidentally to non-interacting static or periodic structures. It is these cells that must be removed, potentially resulting in a splitting of the candidate oscillator. But exactly which cells must be removed is not known until the completion of stage A, and so a second removal stage B is needed.

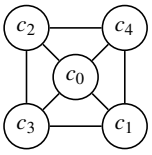


**Fig. 18.98** Oscillator graph $G(O_\pi)$.

#### 18.8.1.2 Removal Stage B

The objective of the second stage is to remove cells from the oscillator that are not required by the oscillator. Given an oscillator, we test any two sub-structures connected by only a dead cell (for example, cell $e$ in Fig. 18.99). We test for the existence of a time step in which one of the structures would cause this connecting dead cell to become live, were it not for presence of the other structure. If there is such a time step, then the dead cell is necessary to the oscillator; if there is no such time step, the dead cell should be removed, and the oscillator split in two.

We perform this removal stage by labelling oscillator cells and considering oscillator neighbourhood graphs.

Define $G(C)$ to be the (possibly disconnected) graph corresponding to a set of cells $C$, which has a node for every $c \in C$ and an undirected edge between every $c_n, c_m \in C$ if $c_n \in N(c_m)$. For an oscillator $O$ we take $G(O)$ to be the oscillator graph of all the cells in $O$. For example, the plinker $O_\pi$ has the oscillator graph shown in Fig. 18.98.

Define a $\gamma$-cell to be a cell $c$ in an oscillator $O$ that remains dead for every timestep $i$ of the oscillation: $\forall t \bullet \sigma_t(c) = \square$.

Define $\gamma(O)$ to be the set of all $\gamma$-cells in $O$: $\gamma(O) = \{\, c \in O \mid \forall t \bullet \sigma_t(c) = \square \,\}$. Define the complement, $\hat{\gamma}(O)$, to be the set of all cells in $O$ that are not $\gamma$-cells (the set of all cells that are alive for at least one timestep): $\hat{\gamma}(O) = \{\, c \in O - \gamma(O) \,\}$. Note that $\gamma(O)$ and $\hat{\gamma}(O)$ partition the cells in $O$.

So $G(\hat{\gamma}(O))$ is the (potentially disconnected) graph of all cells in $O$ that are not $\gamma$-cells (the graph of all cells that are alive for at least one timestep). $G(\hat{\gamma}(O))$ will be a disconnected graph if there are sub-graphs in the oscillator that are disconnected but for a $\gamma$-cell.

Define $G_\kappa(\hat{\gamma}(O))$ to be the set of the disconnected sub-graphs in $G(\hat{\gamma}(O))$. So the size of the set $G_\kappa(\hat{\gamma}(O))$ is the number of sub-graphs in $G(\hat{\gamma}(O))$. There are no neighbouring nodes in different sub-graphs: if $g_1, g_2 \in G_\kappa(\hat{\gamma}(O))$ then $g_1 \neq g_2 \Leftrightarrow \forall c_1 \in g_1, c_2 \in g_2 \bullet c_2 \notin N(c_1)$.

Define an internal $\gamma$-cell to be a $c_\gamma \in \gamma(O)$ that is connected to only one sub-graph in $G_\kappa(\hat{\gamma}(O))$, and so does not connect otherwise disconnected sub-graphs in $G(\hat{\gamma}(O))$.

For example, consider the potential oscillator $P$ (Fig. 18.99) after detection stage A; Fig. 18.100 shows the oscillator graph $G(P)$. $P$ contains a $p2$ plinker next to a $p1$ still life, positioned such that they share a common neighbour, a $\gamma$-cell. The plinker does not require the still life for its $p2$ oscillation, and the still life does not require

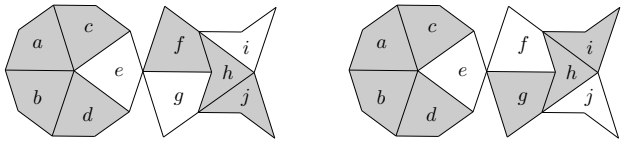


**Fig. 18.99** A potential oscillator $P$ found by detection stage A. It contains a $p2$ plinker with a close stable static structure, and the "on" cells are shown for its two timesteps. Cell $e$ is always off and is the only $\gamma$-cell.

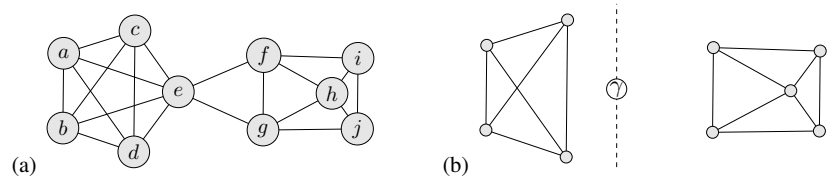


**Fig. 18.100** a) Oscillator graph $G(P)$ marked with cell labels from Fig. 18.99; b) Oscillator graph $G(\hat{\gamma}(P))$; the $\gamma$-cell is not in $G(\hat{\gamma}(P))$ but is drawn for clarity; the dotted line delineates the two sub-graphs. There are two disconnected sub-graphs, so $|G_{\kappa}(\hat{\gamma}(P))| = 2$.

the plinker for its $p1$ oscillation. From the oscillator graph in Fig. 18.100 it is clear that removal of the $\gamma$-cell would disconnect these two independent structures. This is the aim of stage B: to remove unwanted $\gamma$-cells whilst leaving the needed ones.

There are several examples earlier of oscillators that need $\gamma$-cell survival, including disconnected still lifes (Figs. 18.25 and 18.28), the juggler (Fig. 18.85), and the long juggler (Fig. 18.94). In all of these cases, the $\gamma$-cell has exactly three live neighbours in at least one of the subgraphs for at least one timestep. It is stopped from coming alive in the following timestep by live neighbours in another subgraph in this timestep. If this other component were not present (if the oscillator were split into parts), then this cell would become live, and the assumed structure would not be preserved. Hence each of these oscillators is a true disconnected oscillator, and not two separate oscillators in close proximity.

Removal stage B works as follows. Internal $\gamma$-cells are removed[6]. Each remaining $\gamma$-cell connects two otherwise disconnected structures, and is checked for a timestep in which one of the disconnected structures contains exactly three live neighbours of the $\gamma$-cell. If there is such a timestep, then the $\gamma$-cell survives, otherwise it is removed. (Since by definition the $\gamma$-cell is always dead, then if the $\gamma$-cell is connected to a sub-graph containing three live neighbours, then there must be some other sub-graph disconnected from the first with at least one live neighbour of the $\gamma$-cell that prevents the $\gamma$-cell from coming alive. Hence that $\gamma$-cell in needed.) The algorithm

[6] Note that internal $\gamma$-cells can also be necessary for the survival of an oscillator, in that there may be three live cells from one part of the oscillator that would cause the $\gamma$-cell to come alive, were it not for more live cells from another, but still connected, part. The "holes" in the oscillator might need to be "narrow". However, we ignore this point here because we are using the oscillator graph simply to classify oscillators, not to discover suitable patches of tilings to support them.

```
 1: O := O − {internal γ-cells}
 2: {any γ-cells still in O connect sub-graphs}
 3: for each γ-cell c_γ ∈ γ(O) {check for its survival requirement} do
 4:     for each sub-graph g ∈ G_κ(γ̂(O)) do
 5:         for each timestep t do
 6:             if |N_t^■(c_γ) ∩ {c ∈ g}| = 3 { c_γ has three live neighbours at t in g} then
 7:                 mark c_γ for survival
 8:             end if
 9:         end for
10:     end for
11: end for
12: O := O − {all γ-cells not marked for survival}
13: {O now contains only those γ-cells it needs}
14: for each disconnected sub-graph g ∈ G_κ(O) do
15:     construct a new oscillator from g
16: end for
```

**Fig. 18.101** Removal stage B algorithm, for potential oscillator $O$

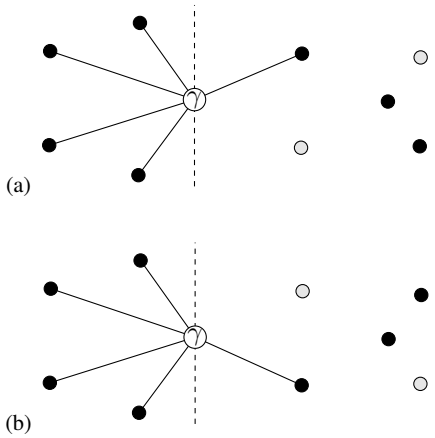


**Fig. 18.102** The live neighbours of the single $\gamma$-cell $e \in \gamma(P)$ at $t = 0$ (a) and $t = 1$ (b). Black nodes are live cells, grey nodes are dead cells. The edges between nodes that are not connected to the $c_\gamma$ are not shown. There is no timestep at which one of the two sub-graphs has three live neighbours of the $\gamma$-cell $e$, and so it does not survive. This removal leaves two disconnected sub-graphs, and hence two resulting oscillators.

for removing unnecessary $\gamma$-cells and constructing the true oscillators from a potential oscillator $O$ is given in Fig. 18.101.

The $\gamma$-cell in potential oscillator $P$ does not meet these survival conditions: see Fig. 18.102.

An oscillator that does require $\gamma$-cells is kd-p3-16-13 (Fig. 18.103). After detection stage A it has seven $\gamma$-cells (Fig. 18.104). Three of these are internal $\gamma$-cells, and so are removed (Fig. 18.105a). The remaining four $\gamma$-cells connect otherwise disconnected sub-graphs (Fig. 18.105b), and must be checked for survival. The checking

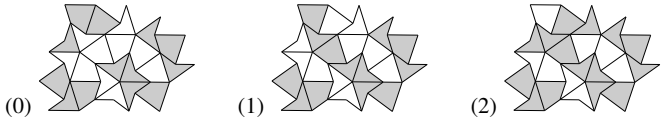


**Fig. 18.103** The example kd-p3-16-13 oscillator (see also Fig. 18.65a), at $t = 0, 1, 2$. Every cell depicted is included in the oscillator after detection stage A.

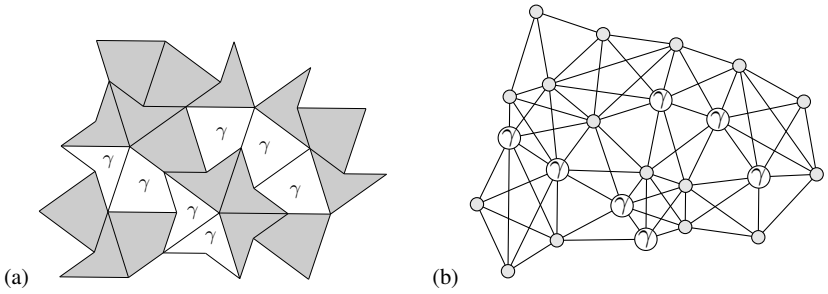


**Fig. 18.104** The example kd-p3-16-13 oscillator: a) the marked $\gamma$-cells; b) graph with marked $\gamma$-cells.

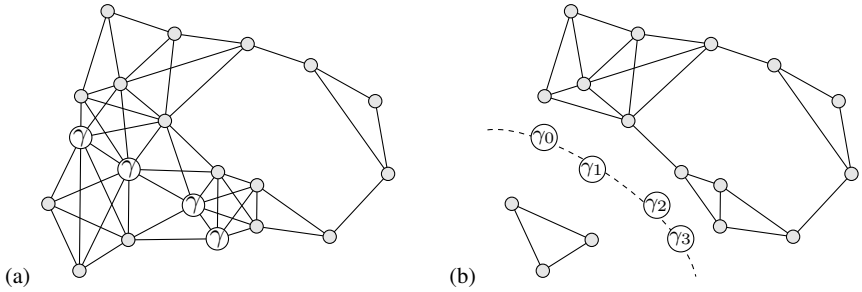


**Fig. 18.105** The example kd-p3-16-13 oscillator: a) the graph with internal $\gamma$-cells removed; b) the disconnected sub-graphs: there are four $\gamma$-cells that require checking.

of the survival conditions is shown in Fig. 18.106: all four $\gamma$-cells meet the survival condition.

The fact that all four $\gamma$-cells survive, each connecting the same two sub-graphs, presents interesting issues. For example, one might wish to eliminate three of the $\gamma$-cells from the description, as only one cell is required to connect the two sub-graphs. Further, one could question whether a tiling variant might produce the same oscillator sub-graphs with a different number of $\gamma$-cells. For the moment we leave the full set of retained $\gamma$-cells in the oscillator description. We discuss these issues further in §18.8.2.2.

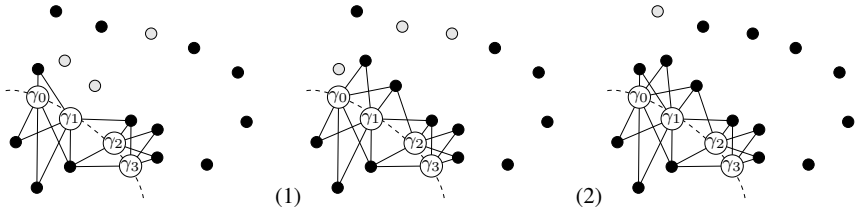


**Fig. 18.106** The example kd-p3-16-13 oscillator, depicting $\gamma$-cell survival conditions at $t = 0, 1, 2$. Live nodes are black, dead nodes are grey, edges between non-gamma cell nodes are not shown. For all time steps the survival condition is fulfilled for $\gamma_0$ and $\gamma_1$, each with three live neighbours in the lower sub-graph. For all timesteps the survival condition is fulfilled for $\gamma_2$ and $\gamma_3$, each with three live neighbours in the upper sub-graph. Note that at $t = 2$ the survival conditions are satisfied for $\gamma_0$ with three live neighbours in the sub-graph also.

## *18.8.2 Oscillator Classification*

### 18.8.2.1 Oscillator Graph Isomorphism

We use the oscillator graph $G(O)$ introduced in §18.8.1.2 in addition to the identification code defined in §18.2.3.1 as the basis for a classification to group the zoo of oscillators into isomorphism classes: those with identical neighbourhood graphs *and* identical codes.

Define two neighbourhood graphs to be identical if they are isomorphic. If $C(O)$ denotes the cells of oscillator $O$, then two oscillators $O_a$, $O_b$ are isomorphic if there exists a mapping $m : C(O_a) \to C(O_b)$ such that any two cells $c_1, c_2 \in O_a$ are neighbours, $c_1 \in N(c_2)$, in $G(O_a)$ if and only if $m(c_1)$ and $m(c_2)$ are neighbours, $m(c_1) \in N(m(c_2))$, in $G(O_b)$. If two oscillator graphs are isomorphic, we write $G(O_a) \sim G(O_b)$.

An oscillator graph isomorphism checking algorithm can use the underlying structure of the neighbourhood graphs, and the limited ways that they can be extended, to make the checks efficient.

The oscillator graph defines the topology of cells and neighbourhoods involved in the oscillator (including any $\gamma$-cells), but does not define which cells are live at each timestep of the oscillator. Note that the oscillator graph alone is not sufficient to uniquely identify an oscillator, although exceptions are rare. Two different oscillators may share a graph; in particular, the $p2$ *fast shuffler* (Fig. 18.48) and the $p4$ *shuffler* (Fig. 18.69) share a graph, Fig. 18.118c.

We have not (at least thus far) found a case where two clearly different oscillators share the same graph *and* the same code, and so we use this combination to identify isomorphic oscillators. If such cases are subsequently discovered, a further disambiguation marking would be needed.

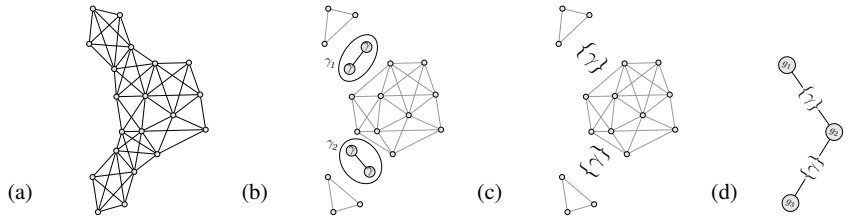


**Fig. 18.107** Constructing the macroscopic graph for $D$, the kd-p5-16-10 *drummer* oscillator (Fig. 18.79): a) the oscillator graph $G(D)$; b) the oscillator sub-graphs $G_\kappa(\hat{\gamma}(D))$ and the $\gamma$-cell sub-graphs $\gamma_1, \gamma_2 \in G_\kappa(\gamma(D))$; c) the oscillator sub-graphs $G_\kappa(\hat{\gamma}(D))$, and markers corresponding to $\gamma_1$ and $\gamma_2$; d) the macroscopic graph $M(D)$.

### 18.8.2.2 Oscillators with $\gamma$-Cells: Macroscopic Isomorphism

As noted at the end of §18.8.1.2, there is a potential issue with the classification of disconnected still lifes. Two oscillators may have isomorphic sub-graphs in $G_\kappa(\hat{\gamma}(O))$ that are connected by differing numbers of $\gamma$-cells: should these be considered the same, of different, oscillators?

We define an extra layer of classification, which allows us to say that such oscillators are *macroscopically* isomorphic, but may be *microscopically* distinct. A macroscopic or microscopic classification may be appropriate in different contexts.

To define macroscopic isomorphism we first define the macroscopic oscillator graph of a disconnected oscillator $O$. Consider the disconnected subgraphs of $\gamma$-cells in $O$, given by $G_\kappa(\gamma(O))$. Each $g_\gamma \in G_\kappa(\gamma(O))$ is a connected graph representing $\gamma$-cells in $O$. Define $M(O)$, the *macroscopic graph* of $O$, to have a node for every $g_i \in G_\kappa(\hat{\gamma}(O))$ and an edge for every $g_\gamma \in G_\kappa(\gamma(O))$ connecting the relevant $g_i$ (connecting the sub-graphs that the corresponding *gamma*-cells connect). Note that $M(O)$ discards the specifics of the $\gamma$-cells, and just asserts that there are $\gamma$-cells which connect two otherwise disconnected sub-graphs containing live nodes. Fig. 18.107 shows the construction of the macroscopic graph for an oscillator with multiple $\gamma$-subgraphs in $G_\kappa(\gamma(O))$.

Define two oscillators $O_a$ and $O_b$ to be *macroscopically isomorphic* if their macroscopic graphs are isomorphic, $M(O_a) \sim M(O_b)$, and each pair of sub-graphs represented by their corresponding nodes are isomorphic.

### 18.8.2.3 Classifying the Still Lifes

Figures 18.108–18.115 show the oscillator graphs for various still lifes. (Note that in a still life oscillator graph with no $\gamma$-nodes, every node must have precisely 2 or 3 neighbours.)

The blocks have fully connected oscillator graphs (Fig. 18.108a and c). The graphs serve to distinguish the 4 cell tubs from the 4 cell blocks (Fig. 18.108b and

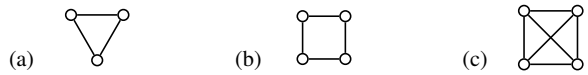


**Fig. 18.108** Three and four node graphs, corresponding to the still lifes p1-3 and p1-4: a) three cell Penrose ace block, Fig. 18.18a, and Q block and D block, Fig. 18.19a; b) regular Life tub, Fig. 18.16, and Penrose tubs, Fig. 18.17; c) regular Life block, Fig. 18.16, and the remaining four cell Penrose blocks in Fig. 18.18, and in Fig. 18.19.

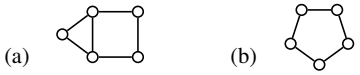


**Fig. 18.109** Five cell graphs, corresponding to the still lifes p1-5: a) regular Life boat, Fig. 18.20; b) five cell kite and dart and rhomb chains, Fig. 18.21.

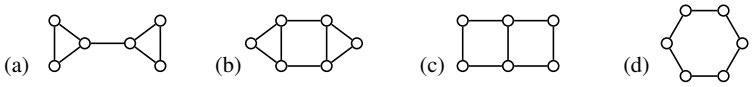


**Fig. 18.110** Six node graphs, corresponding to the still lifes p1-6: a) regular Life snake, Fig. 18.22a and the Penrose 6-snakes, Fig. 18.23; b) the ship; c) the barge; d) the beehive, Fig. 18.22d, and the six cell Penrose chains, Fig. 18.24.

c). This is the case for kite and dart and for rhomb still lifes (and indeed for the regular Life case), justifying the common terminology.

All the discovered five cell chains in Fig. 18.21 have the same oscillator graph (Fig. 18.109b). The regular Life five cell boat has a different oscillator graph (Fig. 18.109a), and so is a distinct structure.

The regular Life snake and Penrose 6-snakes have the same oscillator graph (Fig. 18.110a). The six cell chains have a circular oscillator graph (Fig. 18.110d), as does the regular Life beehive.

The discovered six cell disconnected still lifes exhibit several different kinds of microscopic oscillator graphs (Fig. 18.111a–c). These graphs contain extra nodes corresponding to $\gamma$-cells: cells that are always "dead" but that are necessary to the oscillator; note that each $\gamma$-cell is connected to precisely three nodes in at least one of the "live" subgraphs. Also note that the graphs in Figs. 18.111b and c have isomorphic subgraphs (isomorphic patterns of "on" cells) but different numbers of $\gamma$-cells. They all have the same macroscopic graph (Fig. 18.111d).

The regular Life long snake and Penrose 7-snakes have the same oscillator graph (Fig. 18.112a). All the discovered seven cell rings have the same oscillator graph (Fig. 18.112d), which does not have a chain structure. Rather, it has a main ring of six nodes, with the seventh node providing a third neighbour for two nodes in the main ring. No regular Life still life has this graph; there is a regular Life pattern with a seven node chain: the loaf (Fig. 18.26d).

There are no seven cell regular Life disconnected still lifes. The discovered seven cell Penrose disconnected still lifes exhibit several different kinds of microscopic oscillator graphs (Fig. 18.113a–d). There are two different macroscopic graphs (Fig. 18.113e and f).

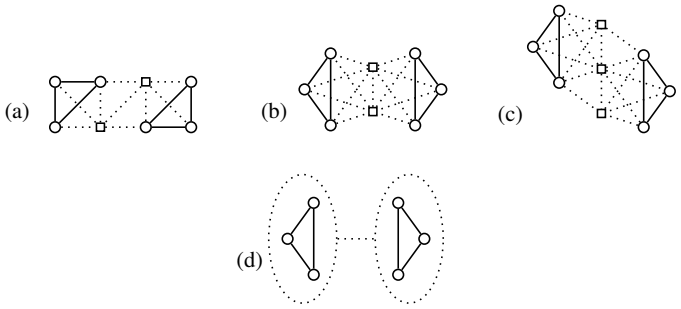


**Fig. 18.111** Six node microscopic graphs for disconnected still lifes p1-6: a) the regular Life carrier in Fig. 18.22; b) the still lifes in Fig. 18.25a and 18.25bi; c) the still lifes in Fig. 18.25bii and iii. In the microscopic graphs, edges from $\gamma$-cell nodes are shown dotted, and $\gamma$-cells are shown as squares, to help clarify the microscopic structure; d) the macroscopic graph for all three forms of still life.

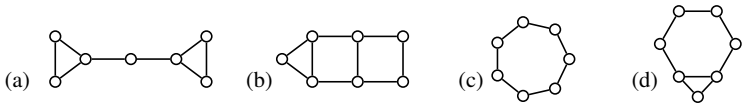


**Fig. 18.112** Seven node graphs, corresponding to the still lifes p1-7: a) regular Life long snake and fishhook, Fig. 18.26, and the Penrose 7-snakes, Fig. 18.27a; b) the long boat; c) the loaf; d) seven cell Penrose rings, Fig. 18.27b–d.

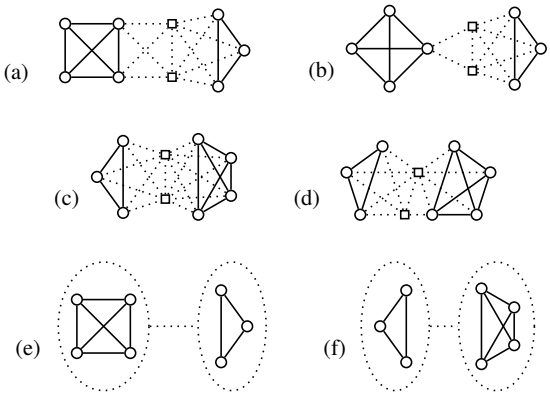


**Fig. 18.113** Seven node microscopic graphs for disconnected still lifes p1-7: a) kd-p1-7, Fig. 18.28a; b) r-p1-7, Fig. 18.28bi; c) r-p1-7, Fig. 18.28bii; d) r-p1-7, Fig. 18.28biii; e) macroscopic graph for a and b; f) macroscopic graph for c and d.

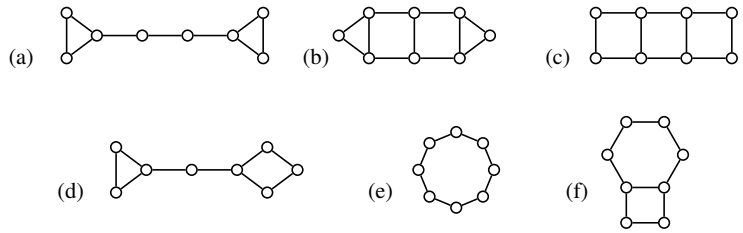


**Fig. 18.114** Eight node graphs, corresponding to the still lifes p1-8: a) regular Life shillelagh, hook with tail, very long snake, and canoe in Fig. 18.29, and the Penrose rhomb 8-snake in Fig. 18.30d; b) the long ship; c) the long barge; d) the tub with tail; e) the pond, the cigar, and the Penrose 8-chains in Fig. 18.30a, bi, c; f) the Penrose S5 vertex variant in Fig. 18.30bii.

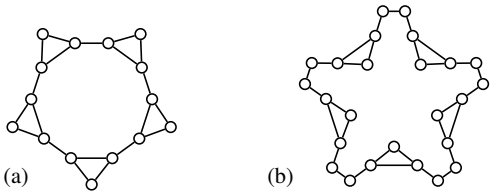


**Fig. 18.115** The graph for the Penrose dart still life rings in a) Fig. 18.35a and Fig. 18.36; b) Fig. 18.35b.

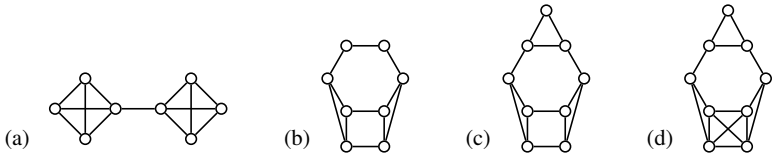


**Fig. 18.116** Graphs for $p2$ oscillators a) l-p2-8-6, the *beacon*, Fig. 18.43b; b) p2-8-6, the *marcher*, Fig. 18.46b and Fig. 18.50b; c) p2-9-7, the *crowned marcher*, Fig. 18.46c and Fig. 18.50c; d) kd-p2-9-6, Fig. 18.46d;

The discovered eight node Penrose still lifes exhibit three different kinds of oscillator graphs: snakes, chains, and rings (Fig. 18.114).

A similar structure to the seven node ring (Fig. 18.112d), but at points all around the main ring, is seen in the dart ring oscillator graphs (Fig. 18.115).

#### 18.8.2.4 Classifying the Oscillators

Figure 18.98 shows the oscillator graph for all kite and dart and rhomb $p2$ plinkers, as well as the regular Life blinker.

Figure 18.116 shows the graphs for the 8 and 9 cell $p2$ oscillators. The regular Life *beacon* (Fig. 18.43b) and the kite and dart *marcher* (Fig. 18.46b) are both p2-8-6, but have different underlying graphs (Figs. 18.116a and b), so are different oscillators. However, the kite and dart marcher (Fig. 18.46b) and the rhomb marcher

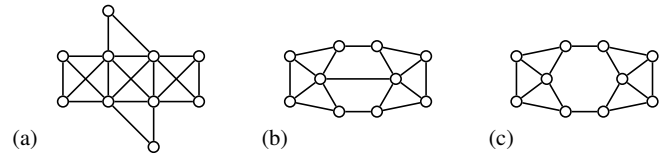


**Fig. 18.117** Graphs for the p2-10-6 oscillators: a) the *toad*, Fig. 18.43c; b) the *clock*, Fig. 18.43d; c) the *hollow clock*, Figs. 18.47 and 18.51.

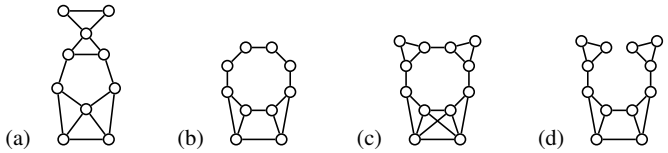


**Fig. 18.118** Graphs for further $p2$ oscillators: a) kd-p2-10-7, Fig. 18.49a; b) r-p2-10-8, Fig. 18.52; c) kd-p2-12-9, isomorphic *fast shuffler*s, Fig. 18.48; and also the graph for the longer period kd-p4-12-10, the *shuffler*, Fig. 18.69; d) r-p2-12-10, isomorphic oscillators from Fig. 18.55b.

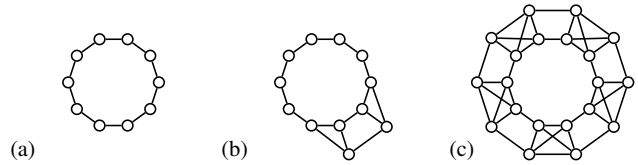


**Fig. 18.119** Graphs for $p2$ oscillators constructed from chains: a) r-p1-10 chain, Fig. 18.61a; b) r-p2-12-10, Figs. 18.55a and 18.61b; c) r-p2-20-10, Fig. 18.62.

(Fig. 18.50b) have the same underlying graph, and so are isomorphic oscillators. Similarly, the p2-9-7 *crowned marcher* (Fig. 18.46c and Fig. 18.50c) are isomorphic.

Figure 18.117 shows the graphs for the p2-10-6 oscillators. The oscillators in Figs. 18.47 and 18.51 have isomorphic graphs (Fig. 18.117c), and are the isomorphic *hollow clock*s. They have similarities to, but are not identical to, the graph of the regular Life *clock* (Fig. 18.117b).

Figure 18.118 shows graphs for further $p2$ oscillators. Note that kd-p2-12-9, the *fast shuffler*, Fig. 18.48 has an isomorphic oscillator graph to the longer period kd-p4-12-10 *shuffler*, Fig. 18.69. Here the code is also needed to distinguish them.

Figure 18.119 shows the graphs for the r-p1-10 chain, and the associated oscillators constructed from it. It demonstrates that the r-p2-12-10 oscillators in Figs. 18.55a and 18.61b are isomorphic to each other, but distinct from the r-p2-12-10 oscillators in Fig. 18.55b, which have a different oscillator graph (Fig. 18.118d).

Figure 18.120 shows the oscillator graphs for the breather, the bat, and the ninja. These further demonstrate how we are able to identify both variant forms (the various kite and dart breathers, and the various bats), and how we are able to identify forms across tilings (the kite and dart breather and the rhomb breather, the kite and dart ninja and the rhomb ninja): they have isomorphic oscillator graphs.

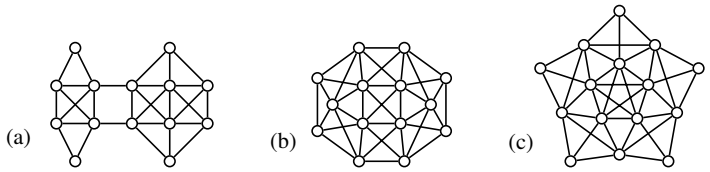


**Fig. 18.120** The graph for a) p3-14-8, the *breather* and all variants, Figs. 18.63 and 18.67; b) kd-p4-14-6, the *bat* and all variants, Fig. 18.71a–c. c) p5-15-8, the *ninja*, Figs. 18.78 and 18.86.

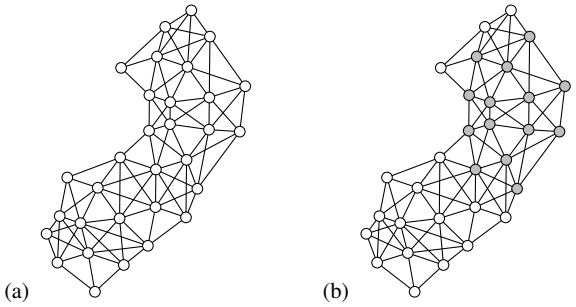


**Fig. 18.121** a) The oscillator graph for kd-p11-31-9, the *malformed bat*, Fig. 18.83; b) the *bat* graph as a subgraph of the *malformed bat* graph.

Note that kite and dart ninja exists on the a0 neighbourhood (Fig. 18.10) rotated around the Sun vertex (Fig. 18.4), and the rhomb ninja exists on the b1 neighbourhood (Fig. 18.12) rotated about the S vertex (Fig. 18.5). The a0 and b1 neighbourhood graphs are isomorphic.

Figure 18.121a shows the oscillator graph for the malformed bat, so named because a part of it has similarities to the bat oscillator. It has the bat oscillator graph as a subgraph (Fig. 18.121b). Having another oscillator as a subgraph is not sufficient to mirror its behaviour, because the graph does not capture which cells are alive at any one time, and the other cells of the oscillator will have some influence on the subgraph cells. Also, due to the structure of the tiling, large oscillators will tend to have subgraphs corresponding to smaller oscillators. Nevertheless, oscillators that do appear visually related will probably have this subgraph structure; for example, the graph of the r-p7-19-7 *hattipper* (Fig. 18.91) appears as a subgraph of the related r-p7-23-8 oscillator (Fig. 18.92).

## 18.9 Summary and Conclusions

We have continued our investigations of the Game of Life on Penrose tilings, producing a preliminary catalogue of still lifes and oscillators, including comparisons with similar oscillators on the regular Life tiling. We have catalogued many still life and periodic oscillators, on both kite and dart and rhomb tilings. We have also

demonstrated that arbitrarily large snakes and chains can exist. We have found no propagating features analogous to gliders, but the existence of "fuses" burning along chains makes us optimistic that such structures may be discovered.

We have introduced an identification code to help refer to different oscillators. However, it does not completely distinguish different oscillators (although collisions are rare, except among the still lifes). There are visually variant forms due to changes in the tiling (for example, the variant oscillators on different $S_C$ extensions); these have the same code. But a few oscillators that are obviously of a different form also share the same code. We further distinguish these by considering the underlying oscillator graph, which captures the topology of oscillator cells neighbourhoods.

The oscillator graph by itself is also insufficient to completely distinguish different oscillators (although collisions are again rare). The combination of code and oscillator graph has, so far, proved sufficient to unify isomorphic forms whilst discriminating different oscillators.

Isomorphic forms can exist on one, two, or all three tilings. For example, several variants of the bats and of dancers exist on the kite and dart tiling; marchers, breathers, and ninjas exist on both kite and dart and rhomb tilings; tubs, blocks, snakes, and blinkers exist on all three tilings. If an oscillator exists on both the regular and one of the Penrose tilings, then it tends to exist on the other Penrose tiling, too. The only exception in this catalogue is that we have not been able to find a kd-p1-8 snake, despite there being a regular Life form (the *very long snake*) and a rhomb form.

Our classification of disconnected oscillators requires the inclusion of $\gamma$-cells: cells that are always dead, but whose removal would destroy the oscillator. We define a further *macroscopic* oscillator graph, to provide a form of isomorphism between disconnected oscillators with different numbers of $\gamma$-cells.

Further work includes extending the catalogue. Regular Life has very many small oscillators; we suspect that our preliminary catalogue of Penrose oscillators has identified only a small proportion of them; a more systematic search is needed. Additionally, larger and longer period oscillators are still to be found. Here an evolutionary search might be more appropriate.

Oscillators themselves, although demonstrating that Penrose life is complex, are not the ultimate goal. That would be to use Penrose life to implement larger computations in a way analogous to regular Life. We have made a start by suggesting that "fuses" can be used to propagate information, or to implement timers. Many more such components are needed: a starting place is to look at the structures and interactions of small oscillators.

## *Acknowledgements*

We thank Adam Nellis for helpful comments on an earlier version.

## References


1. Berlekamp, E.R., Conway, J.H., Guy, R.K.: Winning Ways for Your Mathematical Plays Volume 2: games in particular. Academic Press (1982)
2. de Bruijn, N.G.: Algebraic theory of Penroses non-periodic tilings of the plane I and II. Indagationes Mathematicae (Proceedings) **84**, 39–66 (1981)
3. de Bruijn, N.G.: Remarks on Penrose tilings. In: R.L. Graham, J. Nesetril (eds.) The Mathematics of P. Erdös, volume 2, pp. 264–283. Springer (1996)
4. Gardner, M.: Mathematical games: The fantastic combinations of John Conway's new solitaire game "life". Scientific American **223**(4), 120–123 (1970)
5. Gardner, M.: Mathematical games: extraordinary non-periodic tiling that enriches the theory of tiles. Scientific American **236**(1), 110–121 (1977)
6. Grünbaum, B., Shephard, G.C.: Tilings and Patterns. W. H. Freeman (1987)
7. Hill, M., Stepney, S., Wan, F.: Penrose Life: ash and oscillators. In: M.S. Capcarrere, A.A. Freitas, P.J. Bentley, C.G. Johnson, J. Timmis (eds.) Advances in Artificial Life: ECAL 2005, Canterbury, UK, September 2005, *LNAI*, vol. 3630, pp. 471–480. Springer (2005)
8. Knuth, D.E.: The Art of Computer Programming: Seminumerical Algorithms, volume 2, 3rd edn. Addison Wesley (1998)
9. Niemiec, M.D.: Life page (1998). http://home.interserv.com/˜mniemiec/lifepage.htm
10. Owens, N., Stepney, S.: Investigations of Game of Life cellular automata rules on Penrose tilings: lifetime and ash statistics. In: Automata 2008, Bristol, UK, June 2008, pp. 1–34. Luniver Press (2008)
11. Owens, N., Stepney, S.: Investigations of the Game of Life cellular automata rules on Penrose tilings: lifetime, ash and oscillator statistics. Journal of Cellular Automata **5**(3), 207–225 (2010)
12. Penrose, R.: Pentaplexity. Eureka **39**, 16–32 (1978)
13. Rendell, P.: Turing Universality of the Game of Life. In: A. Adamatzky (ed.) Collision-Based Computing. Springer (2002)
14. Senechal, M.: Quasicrystals and Geometry. Cambridge University Press (1995)
15. Silver, S.: Life lexicon, release 25 (2006). http://www.argentum.freeserve.co.uk/lex.htm
16. Socolar, J.E.S., Steinhardt, P.J.: Quasicrystals. II. Unit-cell configurations. Phys. Rev. B **34**, pp. 617–647 (1986)