96-47

International Journal of Bifurcation and Chaos, Vol. 7, No. 7 (1997) 1451–1496
© World Scientific Publishing Company

COMPLEX ORDER FROM DISORDER AND FROM SIMPLE ORDER IN COARSE-GRAINING INVARIANT ORBITS OF CERTAIN TWO-DIMENSIONAL LINEAR CELLULAR AUTOMATA

ANDRÉ BARBÉ*
Katholieke Universiteit Leuven, Department of Electrical Engineering, Kard. Mercierlaan, 94, B-3001 Leuven, Belgium

Received August 14, 1996; Revised February 19, 1997

This paper considers three-dimensional coarse-graining invariant orbits for two-dimensional linear cellular automata over a finite field, as a nontrivial extension of the two-dimensional coarse-graining invariant orbits for one-dimensional CA that were studied in an earlier paper. These orbits can be found by solving a particular kind of recursive equations (renormalizing equations with rescaling term). The solution starts from some seed that has to be determined first. In contrast with the one-dimensional case, the seed has infinite support in most cases. The way for solving these equations is discussed by means of some examples. Three categories of problems (and solutions) can be distinguished (as opposed to only one in the one-dimensional case). Finally, the morphology of a few coarse-graining invariant orbits is discussed: Complex order (of quasiperiodic type) seems to emerge from random seeds as well as from seeds of simple order (for example, constant or periodic seeds).

1. Introduction

The concept of *coarse-graining invariant orbits* (CGI-orbits) of one-dimensional linear cellular automata (CA) over $\mathbb{Z}_p$ was introduced in [Barbé *et al.*, 1995] and further developed in [Barbé, 1996a]. It concerned two-dimensional orbit patterns which were invariant under coarse-graining renormalization operations. These operations replace each agglomerate of cells in a regular tiling of the orbit configuration space by one new cell whose state is the sum (mod p) of all cell-states in the corresponding agglomerate. Many of these orbits featured structures that were unseen before in the CA-world (and also outside that world). In the last-mentioned reference, a method was presented for finding all solutions of a given CGI-problem under very general problem-settings. An important result was that the bidirectional infinite initial-state configuration for generating the CGI-orbit could be constructed from a *finite seed-configuration* by using proper recursive equations.

In this paper, we will extend that CGI-problem to the case of two-dimensional linear CA over a finite field. This will produce *three-dimensional* coarse-graining invariant structures. It will become clear that this extension to a higher dimension is not a trivial one: Except for a minority of cases, the CGI-solutions generally cannot be obtained from a finite seed. This will also make the nature of the solutions fundamentally different from the one-dimensional counterpart.

*E-mail: Andre.Barbe@esat.kuleuven.ac.be

In Sec. 2, we will formulate the basic CGI-problem and the associated CGI-equations for two-dimensional CA's over the finite field $\mathbb{F}_q$ ($q = p^m$, with p prime and m a positive integer). The elements of $\mathbb{F}_q$ will be denoted by 0, 1, 2, ..., $q-1$. When $m = 1$, then $\mathbb{F}_p \equiv \mathbb{Z}_p$, the field of integers with addition and multiplication modulo p. Depending on the problem parameters, these equations and their solutions will fall into three different categories. These will each be covered, by means of selected examples, in Secs. 3 and 4. Section 5 examines the morphology of some CGI-orbits: It is there that complex order emerges from random seeds as well as from seeds with simple order. For convenience, we will restrict ourselves in Secs. 4 and 5 mainly to binary CA with a linear rule in $\mathbb{Z}_2$, and to simple volume-tilings. This will be sufficient to discuss the fundamental characteristics of the CGI-solutions (and solution methods). Generalizations for renormalizations based on more complex tilings and for the general $\mathbb{F}_{p^m}$ case, as was done in [Barbé, 1996a] for the one-dimensional case, are then rather straighforward extensions of this.

2. Coarse-Graining Invariant Orbits of a Two-Dimensional CA: General Problem Formulation

Orbit decimation

We start here with an overview of some results which follow from a straightforward extension of the one-dimensional case, as presented in [Barbé, 1996a], to the two-dimensional case.

Consider an infinite two-dimensional regular array of cubic cells with natural addresses $(k, l) \in \mathbb{Z} \times \mathbb{Z}$ (see the top layer of cells in Fig. 1). At time $t \in \mathbb{N}$, the cell with address (k, l) is in the state $a(t, k, l) \in \mathbb{F}_q$. This array of cells becomes a CA when it is provided with a dynamics that updates all cell-states synchronously for all times $t \in \mathbb{N}$. The updating of all cells from time t to time $t+1$ is given by the following *local evolution rule*:

$$\forall t \in \mathbb{N} : a(t, k, l) = \sum_{(i,j)\in R_1} \rho(1, i, j)a(t-1, k-i, l-j) \quad (\text{in } \mathbb{F}_q), \tag{1}$$

where R_1 is a finite subset of $\mathbb{Z}^2$, and $\rho(1, i, j) \in \mathbb{F}_q$.

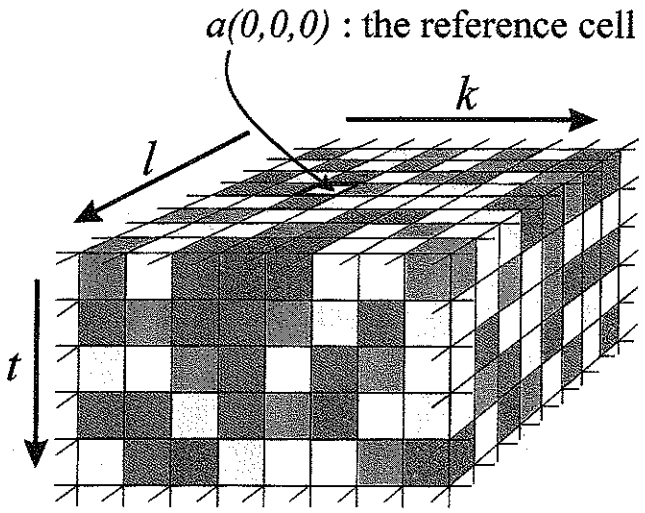


Fig. 1. The orbit configuration space. $a(t, k, l)$ is the state of the cell with address (k, l) at time t: Different states are represented by different colors.

The recursion in Eq. (1) starts at $t = 0$ from an *initial-state configuration* sequence $S_0 \stackrel{\text{def}}{=} (a(0, k, l))_{(k,l)\in\mathbb{Z}^2}$. The sequence $S_t \stackrel{\text{def}}{=} (a(t, k, l))_{(k,l)\in\mathbb{Z}^2}$ is the state of the CA at time t. The state transition rule Eq. (1) can be generalized into

$$\forall t \in \mathbb{N} \text{ and } \forall (t-v) \in \mathbb{N} : a(t, k, l) = \sum_{(r,s)\in R_v} \rho(v, r, s)a(t-v, k-r, l-s), \tag{2}$$

with

$$\rho(v, r, s) = \sum_{(i,j)\in R_1} \rho(1, i, j)\rho(v-1, r-i, s-j) \tag{3}$$

and

$$R_v = \{(r, s) | \rho(v, r, s) \neq 0\}. \tag{4}$$

Associating with S_t the formal two-variable series over $\mathbb{F}_q$

$$S_t(X, Y) = \sum_{(k,l)\in\mathbb{Z}^2} a(t, k, l)X^k Y^l, \tag{5}$$

and associating with $\rho(t, i, j)$ the $\mathbb{F}_q$-polynomial

$$R_t(X, Y) = \sum_{(i,j)\in R_1} \rho(t, i, j)X^i Y^j, \tag{6}$$

[for $t = 1$, we write $R(X, Y)$, which is the *polynomial form* of the local evolution rule] we find that Eqs. (1)–(3) can be written in the equivalent form

$$S_t(X, Y) = S_{t-1}(X, Y)R(X, Y) \tag{7}$$

$$S_t(X, Y) = S_{t-v}(X, Y)R_v(X, Y), \tag{8}$$

with

$$R_v(X, Y) = (R(X, Y))^v . \tag{9}$$

The *orbit* $O_R(S_0)$ of the CA with local rule $R(X, Y)$ and initial state configuration S_0 is given by the three-dimensional sequence $(a(t, k, l))_{(t,k,l)\in\mathbb{N}\times\mathbb{Z}^2}$ generated by continued application of Eq. (1) from a given S_0. Associated with an orbit is a *graphical representation* which attributes a state-dependent color to the cells in a cubic regular lattice defined over $\mathbb{N}\times\mathbb{Z}^2$. This lattice of cells will be called the *orbit configuration space.* The axis orientation is shown in Fig. 1.

Definition 1. Let $I[a, b]$ with $a, b \in \mathbb{Z}$ and $a \le b$ denote the *integer interval* $\{a, a+1, \ldots, b\}$. Denote $I[0, b-1]$ as $\mathcal{Z}_b$. Then define, for fixed t_0, k_0, $l_0 \in \mathcal{Z}_{q^n}$, the (t_0, k_0, l_0)-q^n *decimation* of the orbit $O_R(S_0) = (a(t, k, l))_{(t,k,l)\in\mathbb{N}\times\mathbb{Z}^2}$ as the sequence

$$(d^{(t_0,k_0,l_0)}(t, k, l))_{(t,k,l)\in\mathbb{N}\times\mathbb{Z}^2}$$
$$\stackrel{\text{def}}{=}(a(t_0 + tq^n, k_0 + kq^n, l_0 + lq^n))_{(t,k,l)\in\mathbb{N}\times\mathbb{Z}^2} . \tag{10}$$

Theorem 1. *Any* (t_0, k_0, l_0)-q^n *decimated orbit* $(d^{(t_0,k_0,l_0)}(t, k, l))_{(t,k,l)\in\mathbb{N}\times\mathbb{Z}^2}$ *forms a new orbit satisfying the same rule* R *as the original orbit. Formally:*

$$d^{(t_0,k_0,l_0)}(t, k, l) = \sum_{(i,j)\in R_v} [\rho(v, i, j) \times d^{(t_0,k_0,l_0)}(t - v, k - i, l - j)] . \tag{11}$$

Proof. From Eq. (2), it follows that

$$a(t_0 + tq^n, k_0 + kq^n, l_0 + lq^n) = \sum_{(r,s)\in R_{vq^n}} \rho(vq^n, r, s)a(t_0 + (t - v)q^n, k_0 + kq^n - r, l_0 + lq^n - s) , \tag{12}$$

with $R_{vq^n} = \{(i, j)|\rho(vq^n, i, j) \neq 0\}$, i.e. the set of exponents (i, j) in $R_{vq^n}(X, Y) = \sum \rho(vq^n, i, j)X^iY^j$ for which $\rho(vq^n, i, j) \neq 0$. Because $\mathbb{F}_q$ is a field, $R(X, Y)$ satisfies the *Fermat-property* [Allouche *et al.*, 1996], i.e. $[R(X, Y)]^z = R(X^z, Y^z)$ for $z \in \mathbb{N}$, yielding

$$\begin{aligned} R_{vq^n}(X, Y) &\stackrel{(9)}{=} [R(X, Y)]^{vq^n} \\ &= [R(X^{q^n}, Y^{q^n})]^v \\ &= \sum_{(i,j)\in R_v} \rho(v, i, j)X^{iq^n}Y^{jq^n} \end{aligned}$$

and the right side of Eq. (12) can be written as

$$\begin{aligned} &\sum_{(i,j)\in R_v} \rho(vq^n, iq^n, jq^n)a(t_0 + (t - v)q^n, \\ &\quad k_0 + (k - i)q^n, l_0 + (l - j)q^n) \\ &= \sum_{(i,j)\in R_v} \rho(v, i, j)d^{(t_0,k_0,l_0)}(t - v, k - i, l - j) . \end{aligned}$$

■

Volume-tilings of the orbit configuration space

A q^n-*tile* of the orbit configuration space is any agglomerate of $q^n \times q^n \times q^n$ cells which, by horizontal shifts (uq^n, vq^n) where (u, v) ranges over $\mathbb{Z}^2$, and by vertical shifts wq^n $(w \in \mathbb{N})$, completely covers this space. Such a tile consists of q^n *consecutive* horizontal layers, each of which contains $q^n \times q^n$ cells that are distributed in a certain way. This distribution may depend on the layer, as given by a certain parameter set Ξ, which determines the shape of the tile. With $Q \subset \mathcal{Z}_{q^n} = I[0, q^n - 1]$ and $\overline{Q} = \mathcal{Z}_{q^n}\backslash Q$ two complementary sets of the q^n horizontal layers in a tile, define Ξ as the set of integers

$$\Xi \stackrel{\text{def}}{=} \{\beta_r, \gamma_r, \mu_{r,i}, \nu_{r,i,j}, \beta'_{r'}, \gamma'_{r'}, \mu'_{r',i,j}, \nu'_{r',j}| i, j \in \mathcal{Z}_{q^n}, r \in Q, r' \in \overline{Q}\} ,$$

with $\beta_0 = 0$ (or $\beta'_0 = 0$).

Let $t \in \mathbb{N}$ and $(k, l) \in \mathbb{Z}^2$, then a q^n-tile consists of the set of cells with addresses in a set

$$\begin{aligned} \mathcal{A}_\Xi(tq^n, k, l) = &\{(tq^n + r, k + \beta_r + i + \mu_{r,i}q^n, l + \gamma_r + j + \nu_{r,i,j}q^n)|i, j \in \mathcal{Z}_{q^n}, r \in Q\} \\ &\cup \{(tq^n + r, k + \beta'_r + i + \mu'_{r,i,j}q^n, l + \gamma'_r + j + \nu'_{r,j}q^n)|i, j \in \mathcal{Z}_{q^n}, r \in \overline{Q}\} . \end{aligned} \tag{13}$$

When all parameters in Ξ are 0, the tile has the shape of a cube. The need to distinguish between two subsets in Eq. (13) only exists when one of the μ- or ν-parameters differs from 0, which creates gaps between certain tile-cells (through cell-shifting over distances which are a multiple of q^n). The first subset in Eq. (13) concerns cell displacements in the k-direction first, then in the l-direction; for the second subset, this priority changes. A tile will be identified with this address-set, and so we will speak about the $\mathcal{A}_\Xi(tq^n, k, l)$-tile. Given this tile, any triple $(r, i, j) \in \mathcal{Z}_{q^n}^3$ as it appears in Eq. (13) is the *relative address* of a cell in that tile. The corresponding *absolute address* of a cell with relative address (r, i, j) [this is the full address in Eq. (13)] will be denoted by $c_{(tq^n,k,l)}(r, i, j)$.

The *basis-tile* is the tile $\mathcal{A}_\Xi(0, 0, 0)$. Figure 2 shows a few examples of basis-tiles for $q^n = 2^1$.

The *basic q^n-tiling* $\mathcal{T}_{q^n,\Xi}^{(0,0)}$ of the orbit configuration space consists of the set of tiles that contains the basis-tile:

$$\mathcal{T}_{q^n,\Xi}^{(0,0)} = \{\mathcal{A}_\Xi(tq^n, uq^n, vq^n) | (t, u, v) \in \mathbb{N} \times \mathbb{Z}^2\}.$$

Given a specific basis-tile, there are q^{2n} different tilings of the orbit configuration space, each obtained by a horizontal shift of the tiles over a distance $(\sigma_1, \sigma_2) \in \mathcal{Z}_{q^n}^2$. Denote the corresponding tiling as

$$\mathcal{T}_{q^n,\Xi}^{(\sigma_1,\sigma_2)} = \{\mathcal{A}_\Xi(tq^n, \sigma_1 + uq^n, \sigma_2 + vq^n) | (t, u, v) \in \mathbb{N} \times \mathbb{Z}^2\}.$$

Now, fix a tiling by selecting (σ_1, σ_2). Then the tile $\mathcal{A}_\Xi(tq^n, \sigma_1 + uq^n, \sigma_2 + vq^n)$ is said to have *address* (t, u, v) in that particular $\mathcal{T}_{q^n,\Xi}^{(\sigma_1,\sigma_2)}$-tiling.

Coarse-graining the orbit

Given a CA-orbit $O_R(S_0)$ covered by a tiling $\mathcal{T}_{q^n,\Xi}^{(\sigma_1,\sigma_2)}$, associate with each tile in it a number which is a given linear combination (in $\mathbb{F}_q$) of the states of all cells in the tile. This produces the following mapping of tile-addresses into $\mathbb{F}_q$-numbers:

$$(t, u, v) \to q_W^{(\sigma_1,\sigma_2)}(t, u, v) = \sum_{r,i,j \in \mathcal{Z}_{q^n}} w(r, i, j) a(c_{(tq^n,\sigma_1+uq^n,\sigma_2+vq^n)}(r, i, j)), \quad (14)$$

where $w(r, i, j) \in \mathbb{F}_q$. The three-dimensional matrix $W = [w(r, i, j)]$ of size $q^n \times q^n \times q^n$ is called the *weight-matrix*.

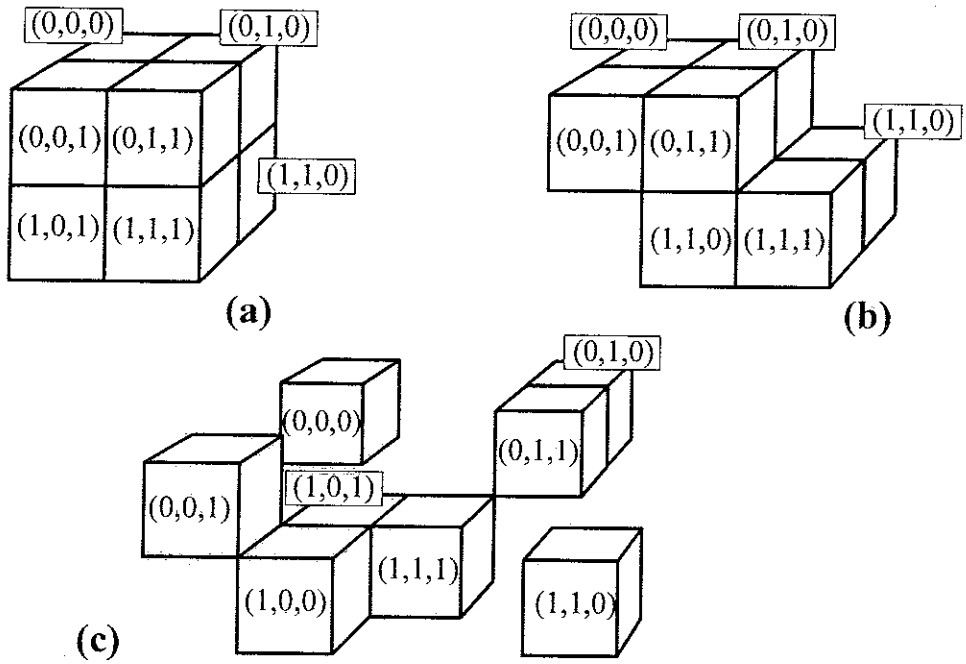


Fig. 2. Three possible $2 \times 2 \times 2$ basis-tiles [see Eq. (13)]: (a) a cubic tile (parameters $\Xi = 0$); (b) $\beta_1 = 1$ is the only Ξ-parameter that differs from 0; (c) $Q = \{0\}$, $\overline{Q} = \{1\}$, $\beta_0 = 0$, $\beta_1' = 1$, $\gamma = 0$, $\gamma_1' = 1$, $\mu_{0,0} = 0$, $\mu_{0,1} = 1$, $\nu_{0,0,0} = 0$, $\nu_{0,0,1} = 1$, $\nu_{0,1,0} = 0$, $\nu_{0,1,1,} = 0$, $\nu_{1,0}' = 1$, $\nu_{1,1}' = 0$, $\mu_{1,0,0,}' = 0$, $\mu_{1,1,0}' = 0$, $\mu_{1,0,1}' = 0$, $\mu_{1,1,1}' = 1$. The cell coordinates in the tile are the (t, k, l)-values in Eq. (13).

The operation which transforms the orbit $O_R(S_0)$ into the sequence $(q_W^{(\sigma_1,\sigma_2)}(t, u, v))_{(t,u,v)\in\mathbb{N}\times\mathbb{Z}^2}$ is called a *coarse-graining transformation* of this orbit and will also be denoted by $[T_{q^n,\Xi}^{(\sigma_1,\sigma_2)}, W]O_R(S_0)$. It has a similar graphical representation associated with it as the original orbit.

Again, in analogy with the one-dimensional case [Barbé, 1996a], the following theorem can be proven as a direct consequence of the fact that a decimated orbit remains an orbit (Theorem 1) and of the linearity of the CA.

Theorem 2. *$[T_{q^n,\Xi}^{(\sigma_1,\sigma_2)}, W]O_R(S_0)$ is a CA-orbit with the same local evolution rule R as the one which generates the original orbit $O_R(S_0)$.*

Notation 1. Let $O_R(S_0) = (a(t, k, l))_{(t,k,l)\in\mathbb{N}\times\mathbb{Z}^2}$ be a CA-orbit and fix some $(c, d) \in \mathbb{Z}^2$. Then ${}^{(c,d)}O_R(S_0)$ denotes the same orbit, but shifted over c cells along the negative k-direction, and over d cells along the negative l-direction, i.e. ${}^{(c,d)}O_R(S_0) = (a(t, k+c, l+d))_{(t,k,l)\in\mathbb{N}\times\mathbb{Z}^2} = O_R({}^{(c,d)}S_0)$. The last symbol denotes the orbit obtained from a (c, d)-shifted version of the initial state configuration S_0.

Theorem 2 allows the formulation of the central problem that follows now.

The coarse-graining invariant problem

Given a two-dimensional CA with local evolution rule $R(X, Y)$, the q^n-tilings $\mathcal{T}_{q^n,\Xi}^{(\sigma_1,\sigma_2)}$ (q^n, Ξ fixed), and a fixed weight matrix W, does there exist an orbit that is invariant under *all* coarse-graining operations $[T_{q^n,\Xi}^{(\sigma_1,\sigma_2)}, W]$? In other words, does there exist an orbit $O_R(S_0)$ such that

$$\forall(\sigma_1, \sigma_2) \in \mathcal{Z}_{q^n}^2 : [T_{q^n,\Xi}^{(\sigma_1,\sigma_2)}, W]O_R(S_0) = {}^{(a_{\sigma_1,\sigma_2},b_{\sigma_1,\sigma_2})}O_R(S_0)\,, \tag{15}$$

where $(a_{\sigma_1,\sigma_2}, b_{\sigma_1,\sigma_2})$ is a shift-vector which depends on (σ_1, σ_2).

Just as for the one-dimensional case, it happens that such CGI-invariant orbits do exist. Finding them requires the solution of Eq. (15).

Coarse-graining invariant equations

Because CA-orbits are completely determined by the initial state configuration, it is necessary and sufficient for Eq. (15) to be true, that the coarse-graining transformation $[T_{q^n,\Xi}^{(\sigma_1,\sigma_2)}, W]$ restricted to the top layer of tiles (i.e. the tiles covering layers 0 to $q^n - 1$ of the orbit), maps this layer into the $(a_{\sigma_1,\sigma_2}, b_{\sigma_1,\sigma_2})$-shifted version of S_0. This produces the *coarse-graining invariant equations*

$\forall(\sigma_1, \sigma_2) \in \mathcal{Z}_{q^n}^2$ and $\forall(u, v) \in \mathbb{Z}^2$:

$$\sum_{(r,i,j)\in\mathcal{Z}_{q^n}^3} w(r, i, j)a(c_{(0,\sigma_1+uq^n,\sigma_2+vq^n)}(r, i, j)) = a(0, u + a_{\sigma_1,\sigma_2}, v + b_{\sigma_1,\sigma_2})\,. \tag{16}$$

Now, it is possible to write the state of the cells on layers 0 to $q^n - 1$ of the orbit [this is the range of r in Eq. (16)], as function of the states in the initial configuration [using Eq. (2)]. Doing so transforms Eq. (16) into

$\forall(\sigma_1, \sigma_2) \in \mathcal{Z}_{q^n}^2$ and $\forall(u, v) \in \mathbb{Z}^2$:

$$\sum_{(i,j)\in\mathcal{N}} \omega(i, j)a(0, \sigma_1 + uq^n + i, \sigma_2 + vq^n + j) = a(0, u + a_{\sigma_1,\sigma_2}, v + b_{\sigma_1,\sigma_2})\,, \tag{17}$$

where $\mathcal{N}$ is a finite subset of $\mathbb{Z}^2$ [dependent on $R(X, Y)$, W, Ξ, but not on (σ_1, σ_2)], and $\omega(i, j) \in \mathbb{F}_q$.

Replacing, for convenience, the initial configuration states $a(0, i, j)$ by $a(i, j)$, Eq. (17) can also be written as

$$\boxed{\begin{array}{l}\forall(\sigma_1, \sigma_2) \in \mathcal{Z}_{q^n}^2 \text{ and } \forall(k, l) : \\ k \equiv \sigma_1 \pmod{q^n}, \quad l \equiv \sigma_2 \pmod{q^n} : \\ \quad \displaystyle\sum_{(i,j)\in\mathcal{N}} \omega(i, j)a(k + i, l + j) \\ \quad = a\left(\dfrac{k - \sigma_1}{q^n} + a_{\sigma_1,\sigma_2}, \dfrac{l - \sigma_2}{q^n} + b_{\sigma_1,\sigma_2}\right).\end{array}} \tag{18}$$

Whatever the value of (k, l), the cell-positions referred to on the left side of this equation are all in the *finite neighborhood* $\mathcal{N}$ of (k, l). Therefore, the left-side terms will be called the (*local*) *renormalizing neighborhood* terms of the CGI-equation. The right-side term will be called the *rescaling* term. We propose to name difference (recurrence) equations of the above type as *renormalizing difference equations*. It is the presence of the rescaling term that makes them basically different from the ordinary difference equations.

When $w(r, i, j) = 0$ for $r = 1, 2, \ldots, q^n - 1$, the left side of Eq. (16) will only contain cell-states of the initial state configuration. In that case, Eq. (16) is identical to Eq. (17), which makes these equations independent of the local evolution rule of the CA. Such CGI-problems and their associated initial configuration solutions will be called *rule-independent.* Observe that Eq. (18) always has a *trivial solution*: the zero-orbit $(a(t, k, l))_{(t,k,l)\in\mathbb{N}\times\mathbb{Z}^2} \equiv 0$. In what follows, it will be shown how to solve the CGI-equations and find all nontrivial solutions. Dependent on the problem parameters (R, W, Ξ), these equations may fall into three categories. Each of them will be dealt with in the coming sections.

3. The Finite Genotype – Infinite Phenotype (FGIP)-Case

It may happen that the left side of the CGI-equation [Eq. (18)] contains only one term. This is the case for orbit invariance under a pure rule-independent q^n-decimation [where the decimation contains cells from the initial state, i.e. $w(r, i, j) = 0$ for all (r, i, j), except for a single element of the form $w(0, i, j)$]. But it may also be the case for CGI-problems which are rule-dependent and not purely decimating. For example, the problem with $q^n = 2^1$, with a pure cubic $2 \times 2 \times 2$-tile (i.e. $\Xi \equiv 0$), with $w(0, 0, 0) = w(0, 1, 0) = w(1, 1, 1) = 1$ and all other elements of the $2 \times 2 \times 2$ W-matrix being zero, and with rule $R(X, Y) = 1 + Y + XY \in \mathbb{Z}_2[X, Y]$, has $a(k + 1, l + 1)$ as unique term in the left-side of the CGI-equation Eq. (18). The CGI-equations are then the same as for the pure 2-decimation case in which $w(0, 1, 1) = 1$ while all other W-elements are 0. In general, with a unique left-side term, in which (i_0, j_0) is the single element of the neighborhood $\mathcal{N}$, the CGI-equation can be written in the following form:

$\forall(\sigma_1, \sigma_2) \in \mathcal{Z}_{q^n}^2$ and $\forall(k, l)$:

$$\omega a(k, l) = a\left(\frac{k - i_0 - \sigma_1}{q^n} + a_{\sigma_1,\sigma_2}, \frac{l - j_0 - \sigma_2}{q^n} + b_{\sigma_1,\sigma_2}\right), \tag{19}$$

where $\sigma_1 = (k - i_0) \bmod q^n$ and $\sigma_2 = (l - j_0) \bmod q^n$, and ω is a coefficient in $\mathbb{F}_q$.

Remark. When $\omega = 1$, (and this is always the case when $q = 2$), there are q trivial solutions, namely $(a(k, l)) \equiv c$, $c \in \mathbb{F}_q$.

Definition 2 [All related to CGI-equations of the type of Eq. (19)]. The *rescaled address* $\mathrm{Res}(k, l)$ of (k, l) in Eq. (19) is the couple of coordinates that appear in the rescaling term:

$$\mathrm{Res}(k, l) = \left(\frac{k - i_0 - \sigma_1}{q^n} + a_{\sigma_1,\sigma_2}, \frac{l - j_0 - \sigma_2}{q^n} + b_{\sigma_1,\sigma_2}\right). \tag{20}$$

This function will also be used componentwise. We then write

$$\mathrm{Res}_{\sigma_2}(k) = \frac{k - i_0 - \sigma_1}{q^n} + a_{\sigma_1,\sigma_2}$$
$$[\sigma_2 \in \mathcal{Z}_{q^n},\ \sigma_1 = (k - i_0) \bmod q^n] \tag{21a}$$

$$\mathrm{Res}_{\sigma_1}(l) = \frac{l - j_0 - \sigma_2}{q^n} + b_{\sigma_1,\sigma_2}$$
$$[\sigma_1 \in \mathcal{Z}_{q^n},\ \sigma_2 = (l - j_0) \bmod q^n]. \tag{21b}$$

The fact that $\mathrm{Res}_{\sigma_2}(k + mq^n) = \mathrm{Res}_{\sigma_2}(k) + m$ allows the construction of a *nonempty* interval $\mathcal{S}_k = I[\underline{k}, \overline{k}]$ such that

$$k \in \mathcal{S}_k \Rightarrow \mathrm{Res}_{\sigma_2}(k) \in \mathcal{S}_k \tag{22a}$$

$$k < \underline{k} \Rightarrow k < \mathrm{Res}_{\sigma_2}(k) \leq \overline{k} \tag{22b}$$

$$k > \overline{k} \Rightarrow k > \mathrm{Res}_{\sigma_2}(k) \geq \underline{k}, \tag{22c}$$

where these properties hold for all $\sigma_2 \in \mathcal{Z}_{q^n}$.

One such interval is given by

$$\underline{k} = \min_{\sigma_1} \left\lceil \frac{q^n \min_{\sigma_2} a_{\sigma_1,\sigma_2} - (\sigma_1 + i_0)}{q^n - 1} \right\rceil \tag{23a}$$

$$\overline{k} = \max_{\sigma_1} \left\lfloor \frac{q^n \max_{\sigma_2} a_{\sigma_1,\sigma_2} - (\sigma_1 + i_0)}{q^n - 1} \right\rfloor. \tag{23b}$$

In a similar way, we can find an interval $\mathcal{S}_l = I[\underline{l}, \overline{l}]$, with

$$\underline{l} = \min_{\sigma_2} \left\lceil \frac{q^n \min_{\sigma_1} b_{\sigma_1,\sigma_2} - (\sigma_2 + j_0)}{q^n - 1} \right\rceil \tag{24a}$$

$$\overline{l} = \max_{\sigma_2} \left\lfloor \frac{q^n \max_{\sigma_1} b_{\sigma_1,\sigma_2} - (\sigma_2 + j_0)}{q^n - 1} \right\rfloor, \tag{24b}$$

that satisfies

$$l \in \mathcal{S}_l \Rightarrow \mathrm{Res}_{\sigma_1}(l) \in \mathcal{S}_l \tag{25a}$$

$$l < \underline{l} \Rightarrow l < \mathrm{Res}_{\sigma_1}(l) \leq \overline{l} \tag{25b}$$

$$l > \overline{l} \Rightarrow l > \mathrm{Res}_{\sigma_1}(l) \geq \underline{l}. \tag{25c}$$

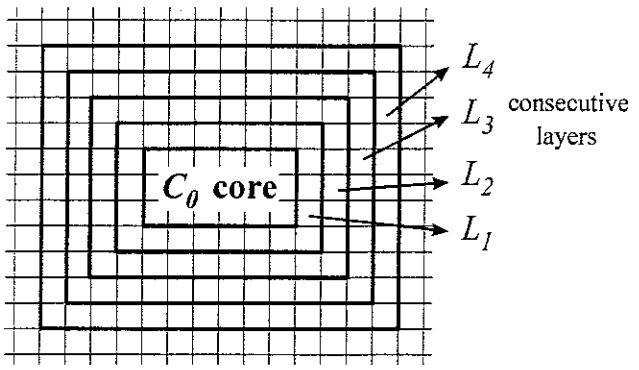


Fig. 3. Structure of the core C_0 and its consecutive layers L_j. $C_j \stackrel{\text{def}}{=} C_0 \cup L_1 \cdots \cup L_j$ is the core-extension up to the jth layer.

$\mathcal{S}_k$ and $\mathcal{S}_l$ as given by Eqs. (23) and (24) are not necessarily minimal, but they can be used as starting points for shrinking them gradually until just before the point where one of the conditions in Eq. (22) or Eq. (25) breaks down. Let $\mathcal{S}_k^{\min} \stackrel{\text{def}}{=} I[\underline{\kappa}, \overline{\kappa}]$ and $\mathcal{S}_l^{\min} \stackrel{\text{def}}{=} I[\underline{\lambda}, \overline{\lambda}]$ denote the *minimal* intervals.

- The *core* (*interval*) is the set of cells with addresses in $\mathcal{C}_0 \stackrel{\text{def}}{=} I[\underline{\kappa}, \overline{\kappa}] \times I[\underline{\lambda}, \overline{\lambda}]$. (This set of cells and the address-set $\mathcal{C}_0$ will be identified).
- Given the core interval $\mathcal{C}_0$, consider a set of *consecutive layers* $\{L_j | j = 1, 2, \ldots\}$ of this core, where the layer L_j is defined as

$$\begin{aligned} L_j \stackrel{\text{def}}{=} & I[\underline{\kappa} - j, \overline{\kappa} + j] \\ & \times I[\underline{\lambda} - j, \overline{\lambda} + j] \backslash I[\underline{\kappa} - j + 1, \overline{\kappa} + j - 1] \\ & \times I[\underline{\lambda} - j + 1, \overline{\lambda} + j - 1] \end{aligned} \tag{26}$$

  (see Fig. 3).
- Call $\mathcal{C}_j \stackrel{\text{def}}{=} \mathcal{C}_0 \cup L_1 \cup L_2 \cdots \cup L_j$ the *core-extension* up to the jth layer.

Theorem 3. *The CGI-equations [Eq. (19)] can be solved, layer by layer, from knowledge of the cell-states in the core interval.*

Proof

(A) We show first that when $(k, l) \in L_j$, then $\text{Res}(k, l) \in \mathcal{C}_{j-1}$.
From Eq. (26), it follows that $(k, l) \in L_j$ means

$$\text{either} \quad (k, l) \in I[\underline{\kappa} - j, \overline{\kappa} + j] \times I[\underline{\lambda} - j, \underline{\lambda} - j] \tag{27a}$$

$$\text{or} \quad (k, l) \in I[\underline{\kappa} - j, \overline{\kappa} + j] \times I[\overline{\lambda} + j, \overline{\lambda} + j] \tag{27b}$$

$$\text{or} \quad (k, l) \in I[\underline{\kappa} - j, \underline{\kappa} - j] \times I[\underline{\lambda} - j + 1, \overline{\lambda} + j - 1] \tag{27c}$$

$$\text{or} \quad (k, l) \in I[\overline{\kappa} + j, \overline{\kappa} + j] \times I[\underline{\lambda} - j + 1, \overline{\lambda} + j - 1]\,. \tag{27d}$$

Consider Eq. (27a). It then follows from Eqs. (22) and (23) that

$$(\underline{\lambda} - j) < \underline{\lambda} \Rightarrow \underline{\lambda} - j + 1 \leq \text{Res}_{\sigma_1}(\underline{\lambda} - j) \leq \overline{\lambda} \tag{28a}$$

$$(\underline{\kappa} - j) < \underline{\kappa} \Rightarrow \underline{\kappa} - j + 1 \leq \text{Res}_{\sigma_2}(\underline{\kappa} - j) \leq \overline{\kappa} \tag{28b}$$

$$(\overline{\kappa} + j) > \overline{\kappa} \Rightarrow \overline{\kappa} + j - 1 \geq \text{Res}_{\sigma_2}(\overline{\kappa} + j) \geq \underline{\kappa}\,. \tag{28c}$$

These last equations imply that for all $i \in I[\underline{\kappa} - j, \overline{\kappa} + j]$:

$$(\text{Res}_{\sigma_2}(\overline{\kappa} + i), \text{Res}_{\sigma_1}(\underline{\lambda} - j)) \in \mathcal{C}_{j-1}\,,$$

meaning that all cells given by Eq. (27a) have their rescaled address in $\mathcal{C}_{j-1}$.

It can be shown in a similar way that $\text{Res}(k, l) \in \mathcal{C}_{j-1}$ for the other cases in Eqs. (27a)–(27d).

(B) What was proven under (A), that $(k, l) \in L_j$ implies that $\text{Res}(k, l) \in \mathcal{C}_{j-1}$, also implies that from knowledge of the cell-states in $\mathcal{C}_{j-1}$, one can determine the cell-states in L_j [through Eq. (19)], and so in $\mathcal{C}_j = \mathcal{C}_{j-1} \cup L_j$. The whole procedure can be recurred starting with $\mathcal{C}_0$. This proves the theorem. ■

So the procedure for solving the CGI-equation [Eq. (19)] seems to be established now: Determine the core, attribute a state to all cells in the core, and get the solution for the initial state configuration of the CGI-solution layer by layer using Eq. (19). Such a core-state configuration acts as a *genotype* (or *seed*) from which the whole infinite solution $(a(k, l))_{(k,l)\in\mathbb{Z}^2}$, the *phenotype*, can be generated. (The terms genotype and phenotype are borrowed from biology, but they are becoming frequently used also in the context of nonbiological complex systems (see for example several papers in [Langton, 1989]).

However, it may be that not all cells in the core may be given arbitrary states in an independent way. There might be equations in the set of CGI-equations [Eq. (19)] whose two terms refer to core-cells [i.e. both (k, l) and $\mathrm{Res}(k, l)$ belong to $\mathcal{C}_0$]. They form the subset of so-called *core-internal equations*, and these have to be solved first. This will in general lead to genotypes in which some cells may be attributed a $\mathbb{F}_q$-state in an arbitrary way (the so-called *freecells*), while the states of the remaining genotype cells will depend on these freecell-states. These cells with a dependent state are called *depcells*. This all seems to be a mere extension of the one-dimensional situation, where all possible solutions to the CGI-problem were generated from a *finite* seed. But, for the two-dimensional situation, this case, where a **F**inite **G**enotype produces the **I**nfinite **P**henotype (the **FGIP**-case), is completely tied to the particular two-term form of the CGI-equations as given in Eq. (19). As such, it forms a minority subset of all possible two-dimensional CGI-problems.

Just like in the one-dimensional case [Barbé, 1996a], it can be shown that all solutions of Eq. (19) form a group isomorphic to the Abelian group $\mathbb{Z}_p^{mN_f}$ (under cellwise addition of the $q = p^m$ states in $\mathbb{F}_q$; N_f is the number of freecells in the seed).

We show here a few examples of solutions of Finite Genotype–Infinite Phenotype CGI-problems. They give but a limited idea of the huge diversity of patterns in which the possible solutions may appear.

Example 1. This example concerns a *rule-independent*, pure 2-decimation invariance: $p^n = 2^1$, cubic $2 \times 2 \times 2$ tile ($\Xi = 0$); $w(0, 1, 0) = 1$, all other $w(r, i, j) = 0$ ($r, i, j \in \{0, 1\}$). $a_{00} = 0$, $b_{00} = -1$, $a_{10} = 1$, $b_{10} = 0$, $a_{01} = 2$, $b_{01} = 0$, $a_{11} = 1$, $b_{11} = 0$. Equation (16) becomes for this case

$$\begin{aligned}
&\forall(u, v) \in \mathbb{Z}^2 : \\
&\quad a(2u+1, 2v) &&= a(u + a_{00}, v + b_{00}) \\
&\quad a(2u+2, 2v) &&= a(u + a_{10}, v + b_{10}) \\
&\quad a(2u+1, 2v+1) &&= a(u + a_{01}, v + b_{01}) \\
&\quad a(2u+2, 2v+1) &&= a(u + a_{11}, v + b_{11}) .
\end{aligned}$$

This can be brought into the form of Eq. (19), with $i_0 = 1$ and $j_0 = 0$. Inserting the actual parameter values gives the CGI-equations

$\forall(k, l) \in \mathbb{Z}^2$:

$$\begin{aligned}
&k = \text{odd} \quad l = \text{even} : a(k, l) = a\left(\frac{k-1}{2}, \frac{l-2}{2}\right) \\
&k = \text{even} \quad l = \text{even} : a(k, l) = a\left(\frac{k}{2}, \frac{l}{2}\right) \\
&k = \text{odd} \quad l = \text{odd} : a(k, l) = a\left(\frac{k+3}{2}, \frac{l-1}{2}\right) \\
&k = \text{even} \quad l = \text{odd} : a(k, l) = a\left(\frac{k}{2}, \frac{l-1}{2}\right) .
\end{aligned} \tag{29}$$

The core-interval is $\mathcal{C}_0 = I[\underline{\kappa} = -1, \overline{\kappa} = 3] \times [\underline{\lambda} = -2, \overline{\lambda} = 0]$. The set of core-internal equations is given by those equations in Eq. (29) for which both terms have their coordinates in the core. Solving this set generates the following seed (genotype) with 5 freecells (these are highlighted):

		k				
		-1	0	1	2	3
	-2	**A**	D	D	**E**	**D**
l	-1	E	D	E	E	**C**
	0	E	**B**	D	D	E

Each of the symbols A, B, C, D, E, can take the value 0 or 1. The complete solution for the initial state configuration can then be obtained through a layer by layer extension of the core using Eq. (29). Figure 4 gives a partial view of 6 solutions out of the $2^5 = 32$ possible ones (the solution display is restricted to the interval $(k, l) \in I[-150, 150] \times I[-150, 150]$). Five of the patterns shown correspond to the cases where only one of the five symbols A, B, C, D, E is given the value 1.

These solutions being pure 2-decimation invariant, deleting every two pixels in a pattern (in both horizontal and vertical direction), and joining the remaining pixels together will produce essentially the same pattern (modulo a shift, and no matter which pixel is selected as the starting pixel).

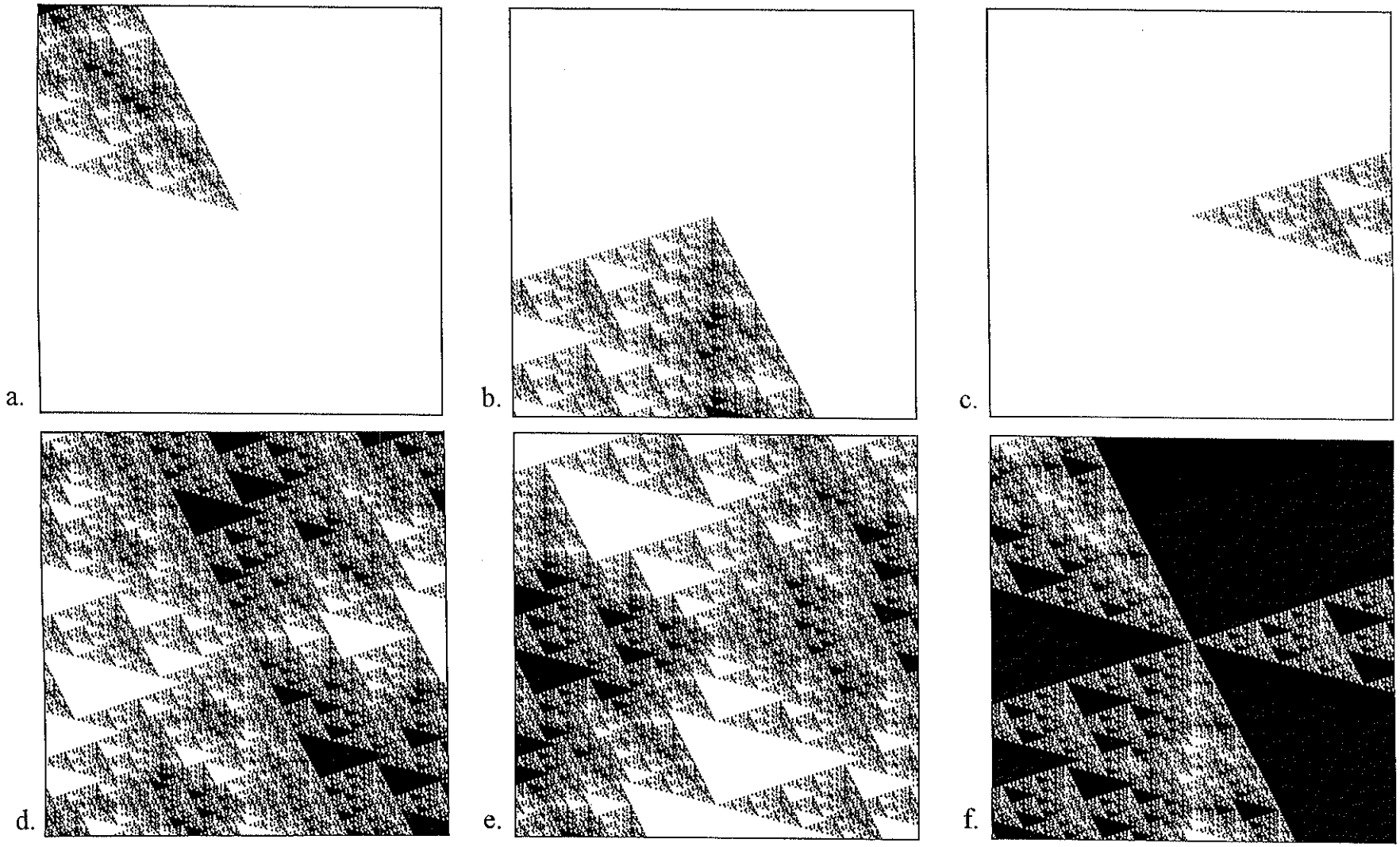


Fig. 4. 6 of the 32 possible initial state solutions for the CGI-problem of Example 1 (partial display over the interval $(k, l) \in I[-150, 150] \times I[-150, 150]$; black pixel = 1, white pixel = 0). Seed configurations: (a) A = 1, B = C = D = E = 0; (b) B = 1, A = C = D = E = 0; (c) C = 1, A = B = D = E = 0; (d) D = 1, A = B = C = E = 0; (e) E = 1, A = B = C = D = 0; (f) D = E = 1, A = B = C = 0.

Each possible solution is the pixelwise addition (modulo 2) of a subset of the set of solutions (a), (b), (c), (d), (e). For example, (f) is the addition of the patterns (d) and (e). These solutions being pure 2-decimation invariant, deleting every two pixels in a pattern (in both horizontal and vertical direction) and joining the remaining pixels, will produce essentially the same pattern (modulo a shift; and no matter which pixel is selected as starting pixel).

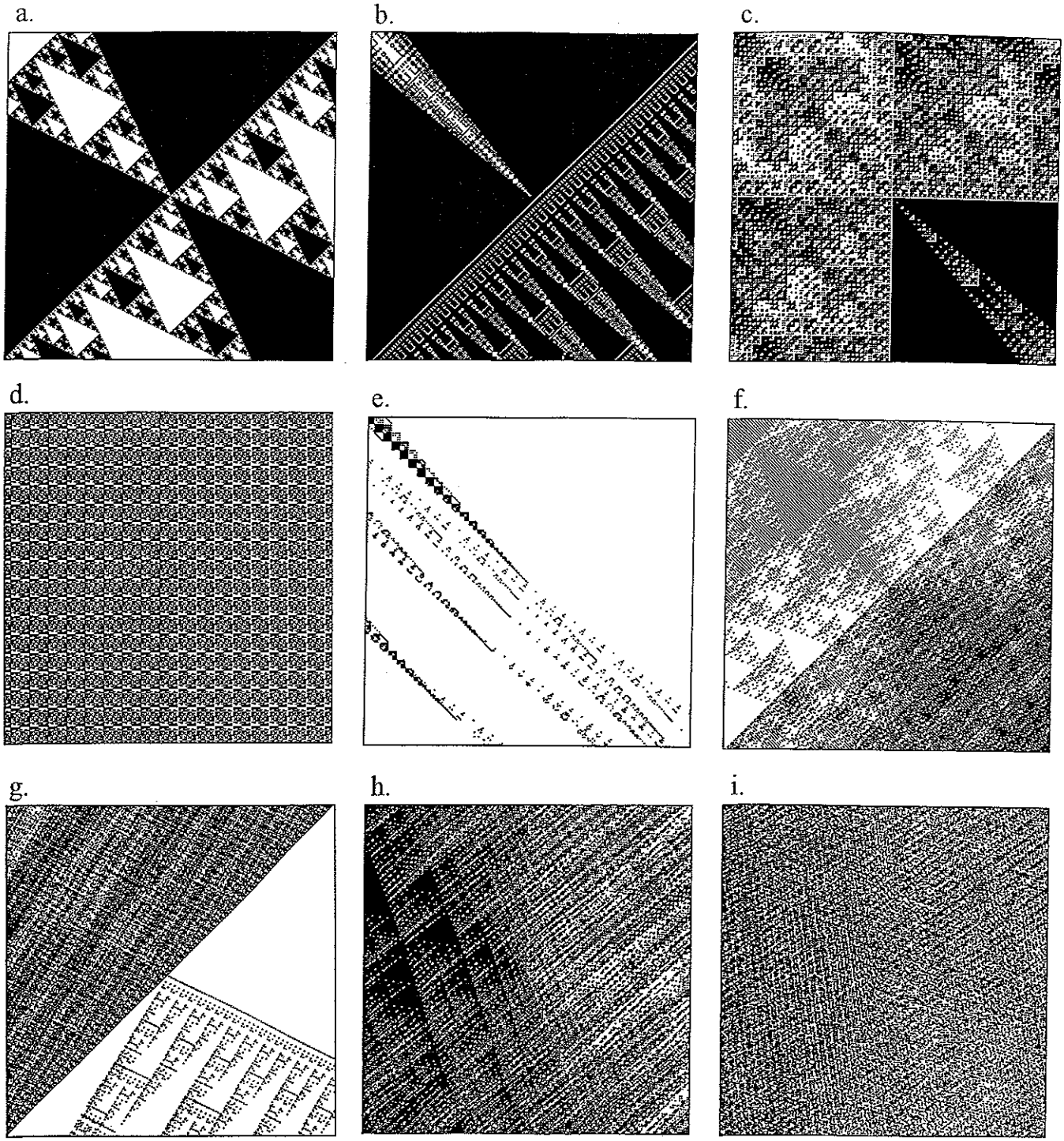


Fig. 5. Additional initial state configurations for the CGI-problem of Example 1, but with different $(a_{\sigma_1,\sigma_2}, b_{\sigma_1,\sigma_2})$-parameters. $[a_{00}, b_{00}, a_{10}, b_{10}, a_{01}, b_{01}, a_{11}, b_{11}]$-parameter configurations: (a) [0, −1, 1, 0, 1, 0, 1, 0]; (b) [0, 0, 2, 2, 2, 2, 0, 0]; (c) [2, 2, 0, 0, 0, 0, 0, 0]; (d) [4, 4, −4, 4, 4, −4, −4, −4]; (e) [0, 0, 4, 4, 8, 8, 0, 0]; (f) [−1, −1, 2, 0, 0, 2, 1, 1]; (g) [1, −2, −1, 2, 0, 0, −2, 2]; (h) [−2, 2, 1, 0, 0, 1, 2, 3]; (i) [2, 3, 1, 4, −2, −1, 2, 0].

Figure 5 shows a few additional CGI-invariant initial configurations that correspond to the problem-settings in Example 1, but with different values for the a_{σ_1,σ_2} and b_{σ_1,σ_2}-parameter set.

Displaying slices from a three-dimensional orbit

In general, we will display part of the initial configuration solutions over the interval $(k, l) \in I[-150, 150] \times I[-150, 150]$ (k in horizontal direction,

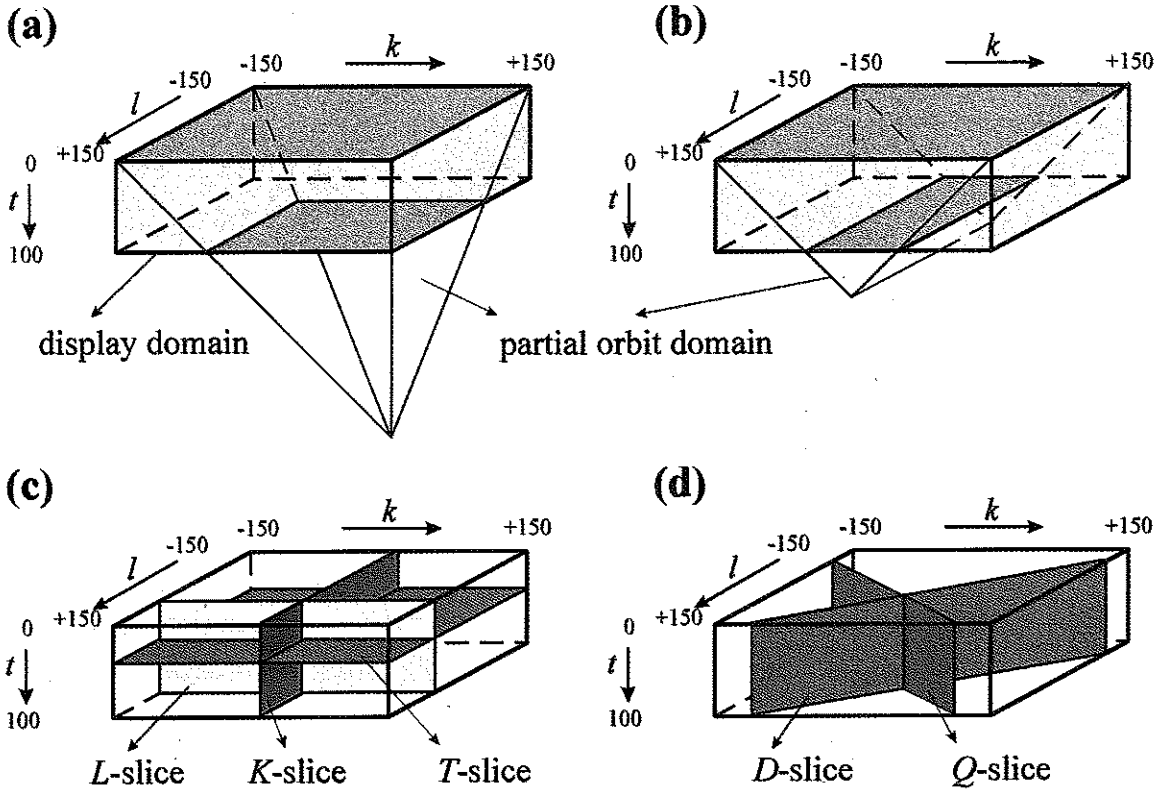


Fig. 6. Three-dimensional *orbit-display domain, partial orbit domain* and two-dimensional *orbit display-slices.*

increasing to the right; l in vertical direction, increasing downwards). From this *partial initial configuration*, only a part of the three-dimensional orbit can be constructed by applying the local evolution rule. This part is confined to some prismoidal or pyramidal domain (top downwards), the exact shape of which is determined by the neighborhood of the rule [see Figs. 6(a) and 6(b)]. It will be called the *partial orbit domain* (associated with the given partial initial configuration).

We will use a three-dimensional *display domain* defined by $(t, k, l) \in I[0, 100] \times I[-150, 150] \times I[-150, 150]$, and will limit the view on the partial orbit within this display. It will only be possible to get an idea about the three-dimensional structure of the orbit, by displaying the structure of a few slices. These slices will be indicated by T_i, L_i, K_i, Q_i, D_i [see Figs. 6(c) and 6(d)]:

T_i: the horizontal slice at $t = i$: $(a(i, k, l))_{(k,l)\in I[-150,150]\times I[-150,150]}$,

L_i: the vertical slice parallel to the k-direction at $l = i$:
$(a(t, k, i))_{(t,k)\in I[0,100]\times I[-150,150]}$,

K_i: the vertical slice parallel to the l-direction at $k = i$:
$(a(t, i, l))_{(t,l)\in I[0,100]\times I[-150,150]}$,

Q_i: the vertical slice parallel to the main diagonal direction in the (k, l)-plane:
$(a(t, k, k+i))_{(t,k)\in I[0,100]\times I[\max(-150,-150-i),\min(150,150-i)]}$,

D_i: the vertical slice parallel to the other diagonal direction in the (k, l)-plane:
$(a(t, k, -k+i))_{(t,k)\in I[0,100]\times I[\max(-150,-150+i),\min(150,150+i)]}$.

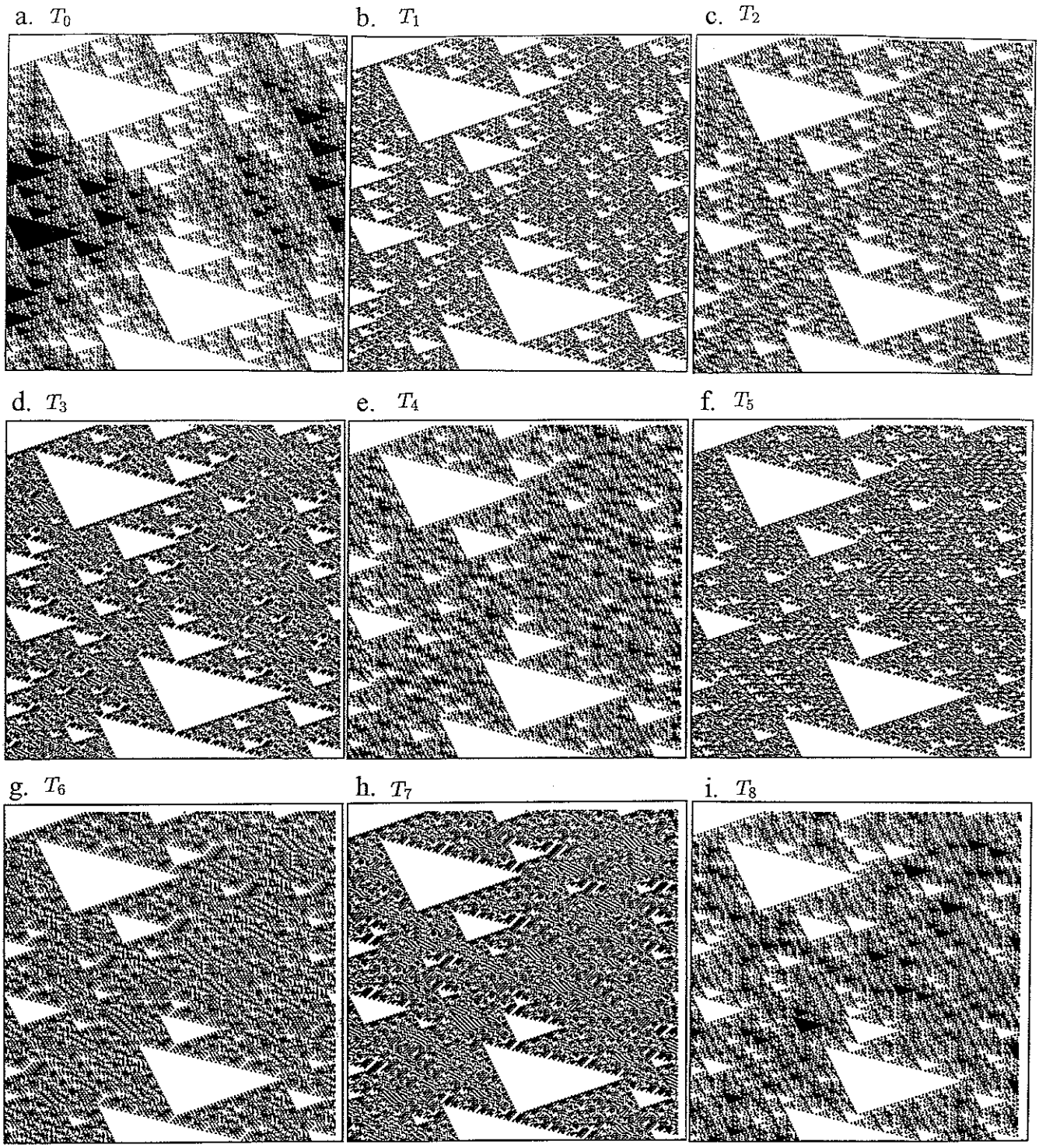


Fig. 7. Orbit slices corresponding to Fig. 4(e). (a)–(i) horizontal slices T_0 to T_8; (j)–(r) T_{24} to T_{32}. Any 2-decimation of T_{2i} reproduces the pattern T_i. So, compare T_1, T_2, T_4, T_8; T_3 and T_6. (s) L_{-10}, L_0, L_{10} (from left to right); (t) K_{-10}, K_0, K_{10}; (u) Q_{-10}, Q_0, Q_{10}; (v) D_{-10}, D_0, D_{10}.

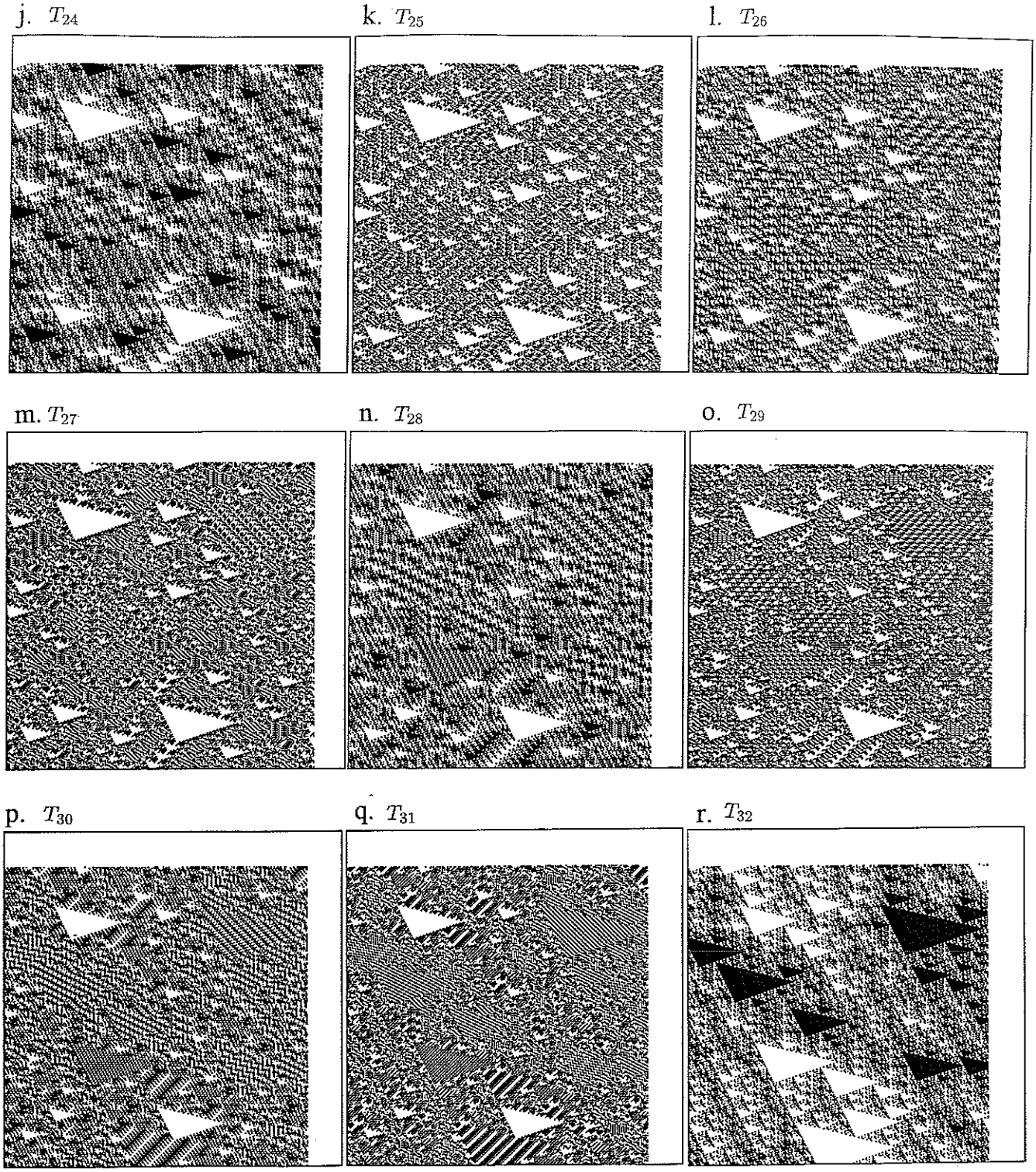


Fig. 7. (*Continued*)

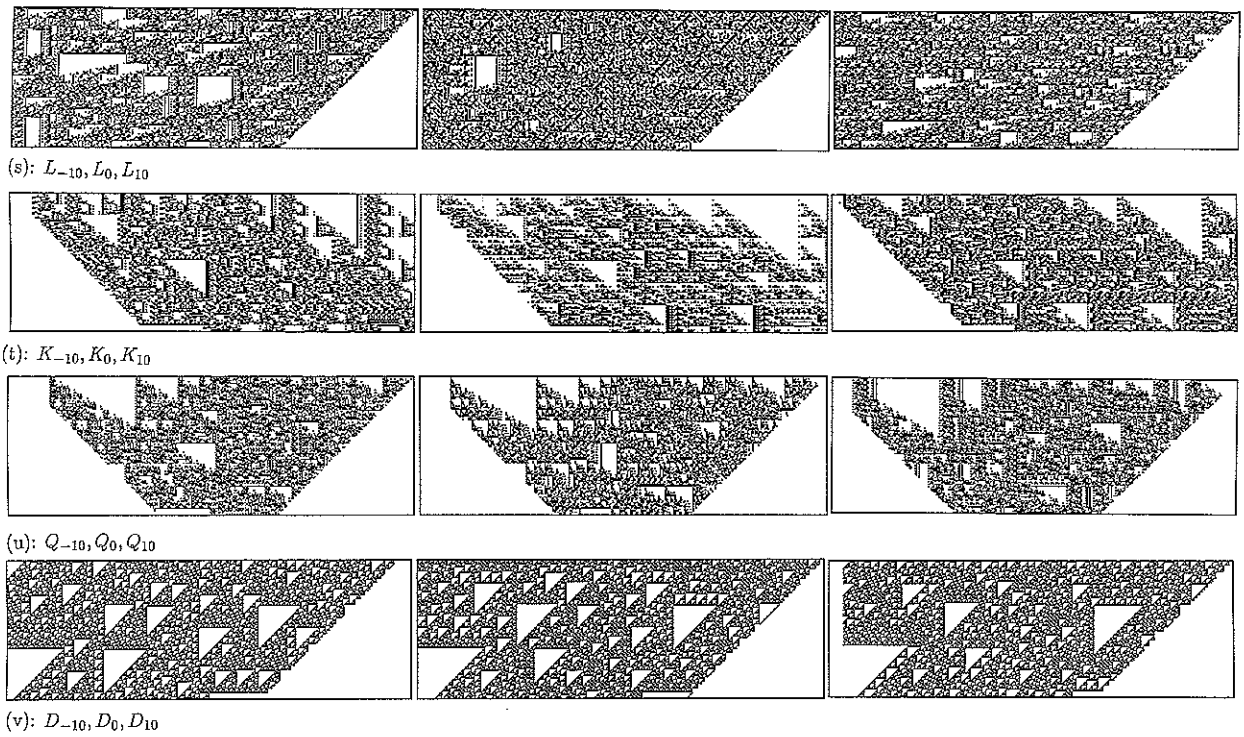


Fig. 7. (*Continued*)

Figure 7 shows a few slices of the orbit obtained from the initial configuration displayed in Fig. 4(e), using the rule $R(X, Y) = 1 + X^{-1}Y$. Observe that, for this particular example, which is a pure 2-decimation invariant solution, any 2-decimation of a T_{2i}-plane, produces the pattern in the T_i-plane (modulo a shift).

Example 2. This is an example for a three-valued automaton over $\mathbb{Z}_3$ ($q^n = 3^1$), with a *rule-dependent* CGI-equation that is of FGIP-type. The basis-tile is a cubic $3 \times 3 \times 3$ tile ($\Xi = 0$). The weight matrix is given by $w(0, 0, 0) = w(0, 0, 1) = w(0, 1, 0) = w(1, 1, 1) = 2$, $w(0, 1, 1) = 1$, and all other $w(r, i, j) = 0$. The rule is $R(X, Y) = 1 + 2X + 2Y + XY \in \mathbb{Z}_3[X, Y]$. The CGI-equations corresponding to Eq. (17) are given by

$\forall(\sigma_1, \sigma_2) \in \{0, 1, 2\}^2$ and $\forall(u, v) \in \mathbb{Z}^2$:

$$a(3u + \sigma_1, 3v + \sigma_2) = a(u + a_{\sigma_1,\sigma_2}, v + b_{\sigma_1,\sigma_2}).$$

These equations can be put into the form of Eq. (19):

$\forall(k, l) \in \mathbb{Z}^2$:

$$a(k, l) = a\left(\frac{k - \sigma_1}{3} + a_{\sigma_1,\sigma_2}, \frac{l - \sigma_2}{3} + b_{\sigma_1,\sigma_2}\right), \tag{30}$$

where $\sigma_1 = k \bmod 3$ and $\sigma_2 = l \bmod 3$.

After solving the core-internal equations, which produces the freecells and the depcells, the complete initial configuration solution can again be constructed by expanding the core layer by layer. Figure 8 shows a solution for 4 different cases of the $(a_{00}, b_{00}, \ldots, a_{22}, b_{22})$-vector. Figure 9 shows a few slices of the orbit obtained from the initial configuration of Fig. 8(a).

Observe that the rule-independent pure 3-decimation invariance problem for which the only nonzero element of the weight-matrix is $w(0, 0, 0) = 1$, while all other parameters remain the same, produces the same CGI-equations [Eq. (30)] and hence the same initial configuration solutions.

4. The Infinite Genotype – Infinite Phenotype (IGIP)-Case

In the most general case, the CGI-equations [Eq. (18)] will have a renormalizing neighborhood $\mathcal{N}$ that consists of more than one cell. It will be shown by a few examples that, in order to construct the infinite initial configuration S_0 that leads to a coarse-graining invariant orbit, one needs an *infinite* genotype.

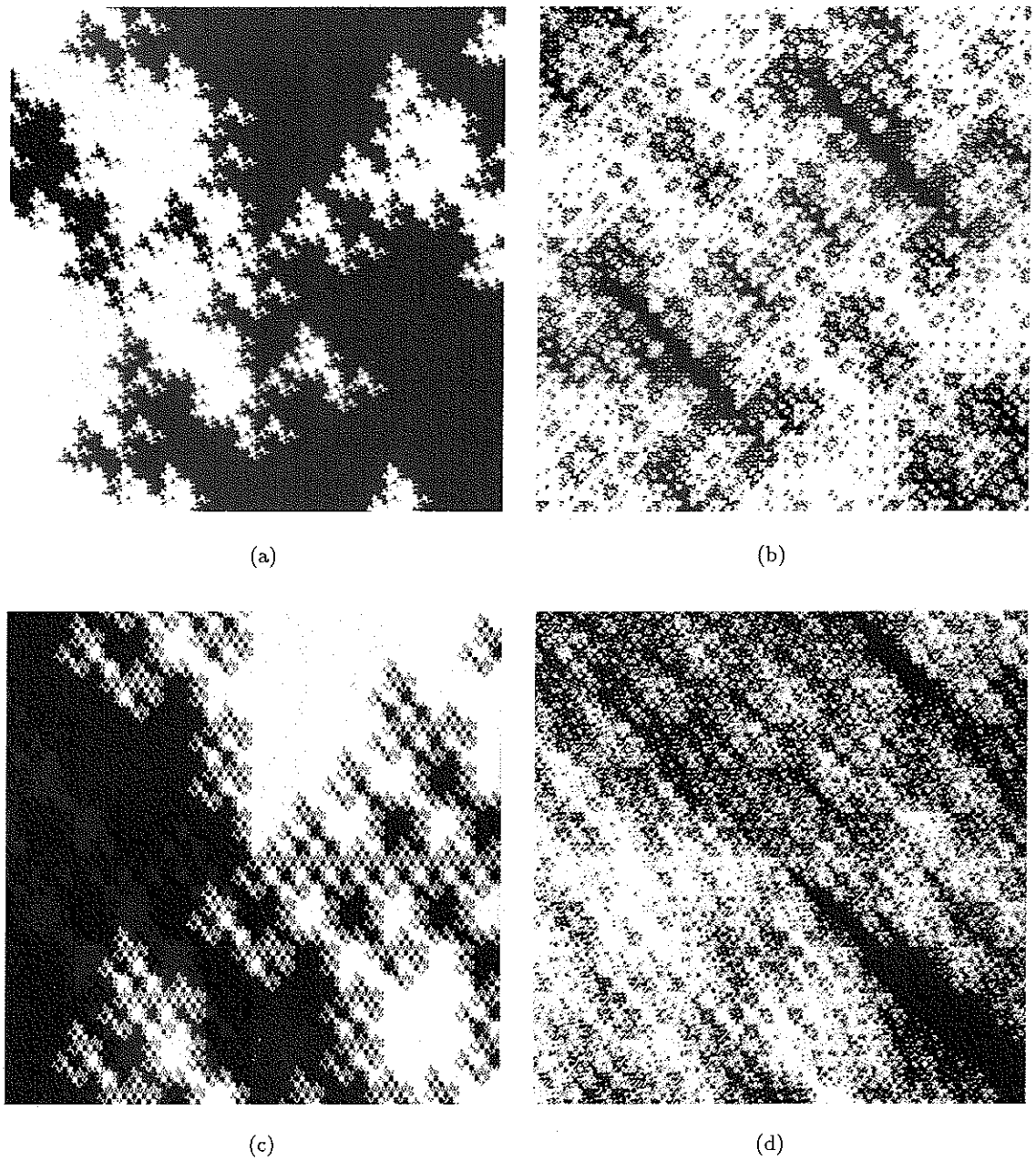


Fig. 8. Initial state configuration solutions for the CGI-problem of a CA in $\mathbb{Z}_3$, as given in Example 2. The $[a_{00}, b_{00}, a_{10}, b_{10}, a_{20}, b_{20}, a_{01}, b_{01}, a_{11}, b_{11}, a_{21}, b_{21}, a_{20}, b_{20}, a_{21}, b_{21}, a_{22}, b_{22}]$-parameter set: (a) [0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]; (b) [3, 3, 0, 0, 0, 0, 0, 0, −2, −2, 0, 0, 0, 0, 0, 0, 1, 1]; (c) [0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1]; (d) [0, 0, 0, 0, 1, 1, 0, 0, 2, 2, 0, 0, 3, 3, 0, 0, 4, 4].

Figure 10 shows a set of neighborhood examples, each associated with a certain tile-configuration, CA-rule and weight-matrix, as given in the Appendix A (just consider the configuration of cells with a black boundary for the time being). It is clear that more complex neighborhoods can be created by invoking more complex tile-configurations and/or CA-rules, but the features of these simple examples are generic for the problem under discussion. Before dealing with the solution of the CGI-equation, it will be useful to consider an incomplete version of it.

4.1. *The CGI-equation without rescaling term, IGIP-C and IGIP-nC classes*

When the rescaling term in Eq. (18) is deleted, we

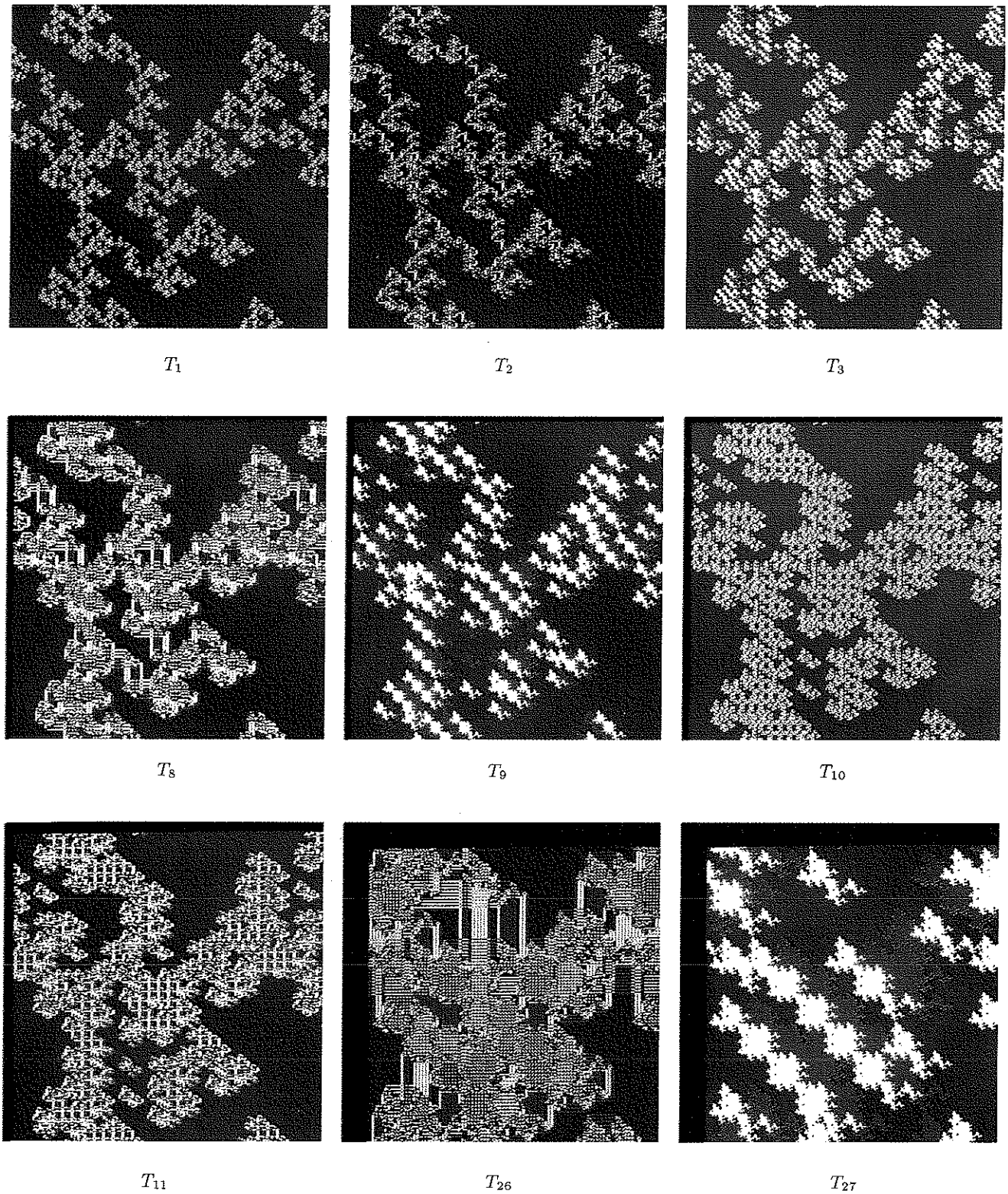


Fig. 9. Some horizontal slices for the orbit generated from Fig. 8(a) by the rule $R(X, Y) = 1 + 2X + 2Y + XY \in \mathbb{Z}_3[X, Y]$. Any 3-decimation of T_{3i} reproduces T_i, so compare T_1, T_3, T_9, T_{27}.

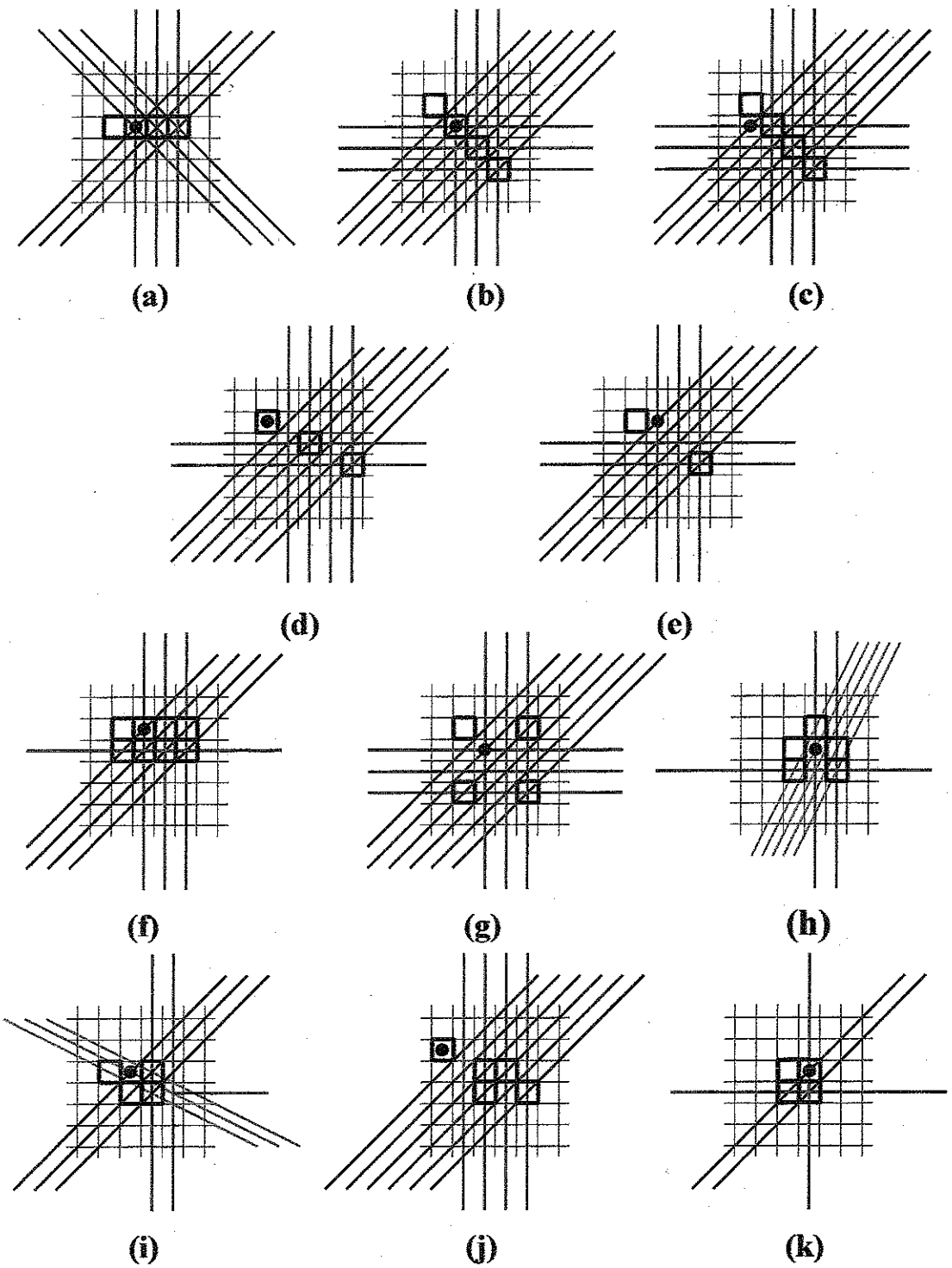


Fig. 10. Examples of different renormalizing neighborhoods of the CGI-equations without rescaling term (the neighborhoods are the sets of black-highlighted cells). The black dot indicates the (0, 0)-reference address for the neighborhood. Each set of lines of the same color forms a set of *core-lines*: A *core* consists of all cells whose centers lie on these core-lines. (a), (b), (c), (d), (e), (j) only *unidirectional* cores; (f), (g), (h), (i), (k) unidirectional and *bidirectional* cores. (a)–(e) *collinear* neighborhoods; (f)–(k) *non-collinear* neighborhoods.

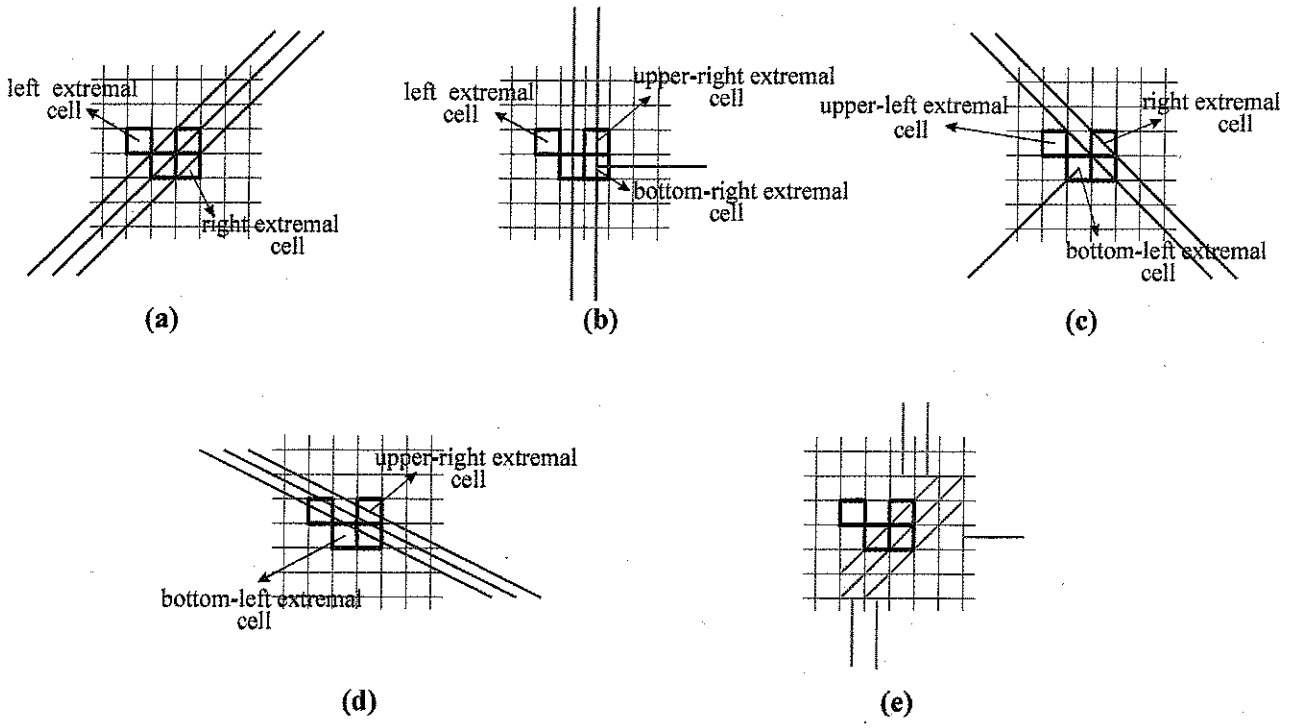


Fig. 11. Different cores and extremal cells for a single neighborhood set. A core is formed by all cells whose center lies on the colored lines (the core-lines). (a), (b), (c), (d) regular cores; (e) part of an irregular core.

obtain an equation of the form

$$\forall(k, l) \in \mathbb{Z}^2 : \sum_{(i,j)\in\mathcal{N}} \omega(i, j)a(k+i, l+j)=0\,. \qquad (31)$$

It says that a particular weighted sum of the CGI-related neighborhood (i.e. a particular renormalization) should be 0, no matter where that neighborhood is located in $\mathbb{Z}^2$. Equation (31) can be solved recursively over $\mathbb{Z}^2$, provided that some well-chosen initial condition (or *seed*) $\mathcal{S}_{\text{wr}}$ is given (wr: without rescaling). Consider for example the neighborhood in Fig. 11(a), then it is sufficient to fix the $a(k, l)$-states along three adjacent diagonal lines, as shown there. Indeed, given this seed, it is clear that the solution can propagate from it layer by layer, either towards the upper-left or to the bottom-right directions, and this is possible in a completely parallel fashion. The neighborhood-cell which is not covered by the already existing solution is called the *extremal cell* [we could be more specific: left-extremal or right-extremal (or upper/bottom extremal), as the extremal cell depends on the direction in which the solution propagates]. As a matter of fact, the solution grows in both directions like the two-dimensional orbit of a *higher-order* one-dimensional cellular automaton, whereby a cell-state at "time t" depends on the states of some neighborhood of this cell over a finite time interval that goes further back in time than $t-1$ (see [Martin *et al.*, 1984]).

The set of seed-cells, or, to be more precise, the set of their associated addresses, forms the *core* that supports the seed. Observe that, just like in the FGIP-case, a seed is a minimal configuration from which the whole solution may develop: when supported by only two of the three core-lines in Fig. 11(a), there is no way to start propagating the solution.

But the core of Fig. 11(a) is not the only possible one: Fixing $a(k, l)$ along two adjacent vertical lines and one horizontal half-line, as shown in Fig. 11(b), would also be a good choice. A third alternative is shown in Fig. 11(c), where $a(k, l)$ is fixed along two diagonal lines and a half-line along the other diagonal direction. Figure 11(d) still gives another possibility, with the cell-states fixed along three *skew* parallel lines.

For obvious reasons, a core like in Figs. 11(a) and 11(d) will be called *unidirectional*; cores like in Figs. 11(b) and 11(c) are *bidirectional*. Observe that a core need not necessarily have a *regular* structure, like the ones just mentioned. Figure 11(e) shows

a part of a more irregular example. Figure 10 shows a few possible regular cores associated with different neighborhoods that may arise in Eq. (31). Notice that it seems always possible to find a unidirectional core, albeit that the direction need not necessarily be horizontal, vertical or strictly diagonal. The examples in Figs. 10(f)–10(k) illustrate this.

Selecting a seed like in Fig. 11(b), only the solution propagating to the left is higher-order CA-like. The solution to the right can only propagate in a pipelined way producing a widening computational front moving along an upper-right direction, and a front moving along the bottom-right direction.

Observe also that, whatever type of core is selected, it can be placed anywhere in the $\mathbb{Z}^2$-plane. This implies that there are actually an infinite number of possible seeds.

Now consider a finite subset of a core and define a seed on it. From this finitely restricted seed, it is only possible to generate a part of the full solution for S_0. The solution domain and the corresponding solution will be called the *restricted core extension domain* (RCED) and the *restricted seed solution* (RSS). Figure 12 shows a few examples of RCED's. Observe that there are two fundamentally different categories. The first category concerns the cases with an *infinite* RCED and associated RSS (which however still forms a proper subset of the complete solution). A necessary and sufficient condition for such a case to happen is that all cells of the neighborhood $\mathcal{N}$ are *collinear*, like in the examples of Figs. 10(a)–10(e). The second category contains the cases whose restricted seed solution is always confined within a *finite* domain, like in the examples of Figs. 10(f)–10(k). Both categories will be commonly labeled "**I***nfinite* **G***enotype*–**I***nfinite* **P***henotype*" (**IGIP**), as opposed to the FGIP-case from Sec. 3. A case in the first category (with

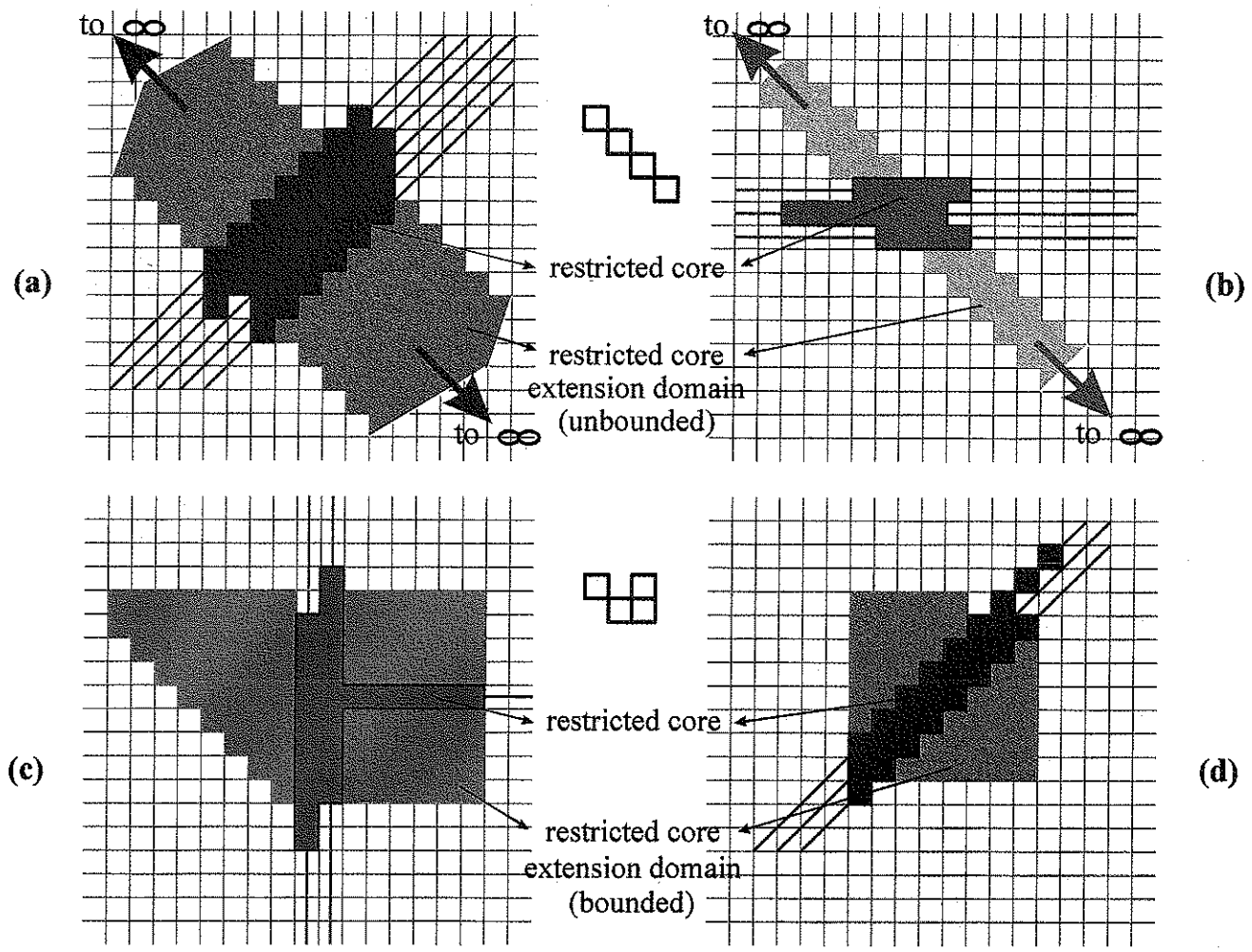


Fig. 12. Restricted cores and restricted core extension domains (RCED's): (a), (b): For the collinear neighborhood of Fig. 10(b): the RCED is unbounded. (c), (d): For the non-collinear neighborhood of Fig. 10(i): the RCED is finite.

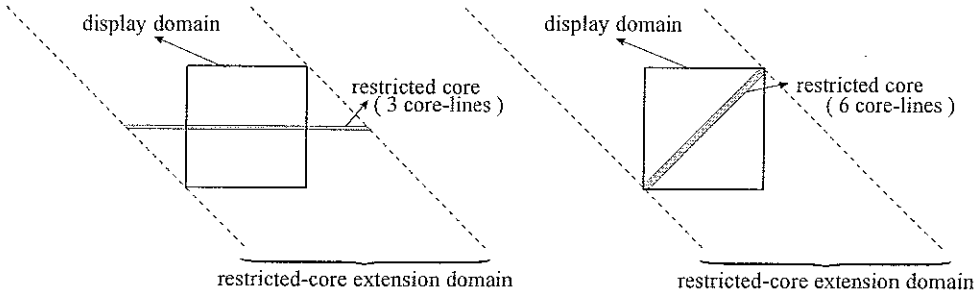


Fig. 13. Different restricted cores with the same extension domain covering the display domain.

collinear neighborhood and infinite RCED) will be classified as an IGIP-C case, one in the second category (with noncollinear neighborhood and finite RCED) as an IGIP-nC case.

In order to display partial solutions of Eq. (31), one should select a restricted seed such that its extension domain completely covers the display domain. Here practical considerations may determine the choice between different possible seeds. This is illustrated in Fig. 13 for the neighborhood in Fig. 10(b). The restricted *horizontal* core must also be partially defined outside the display domain, while this is not so with the *diagonal* core. This is also the reason why there are more core-lines in the diagonal case than in the horizontal case (the number of cells covered by these restricted cores are the same in both cases). Theoretically, there is no difference, and both seeds can in principle be converted to each other so that they produce the same solution.

4.2. *Back to the full CGI-equations*

Let us reintroduce the rescaling term in Eq. (31) to get back the original CGI-equations Eq. (18). In order to be able to solve these CGI-equations recursively, it should be possible at any step in the recursion to find an extremal cell of the neighborhood $\mathcal{N}$ at the outer boundary of an already known part of the solution, such that the nonextremal part of the neighborhood *and* the corresponding rescaling term fall inside the already known solution part. This should apply from the very first step, starting from the seed $\mathcal{S}$. This implies that any seed $\mathcal{S}$ must have a core $\mathcal{C}$ which contains a core $\mathcal{C}_{\mathbf{rw}}$ of the corresponding equation without rescaling term. In most cases, $\mathcal{C}$ will even extend $\mathcal{C}_{\mathbf{rw}}$, because the corresponding rescaling term must also fall inside $\mathcal{C}$ (while it may not necessarily belong to $\mathcal{C}_{\mathbf{rw}}$). As another consequence of the reappearance of the rescaling term, the number of possible cores will become finite.

In order to find a core $\mathcal{C}$ for the CGI-equations, we will start with a corresponding $\mathcal{C}_{\mathbf{rw}}$-structure, and extend this in a trivial way by adding a provisionally indefinite number of extra core-lines adjacent to the core-lines in $\mathcal{C}_{\mathbf{rw}}$. Then the minimal size and exact location of this extended core will be determined by using similar conditions as for finding a core in the one-dimensional CGI-problem as presented in [Barbé, 1996a]. The point is that it is possible to use such conditions in the two-dimensional case for the different coordinates (directions) *separately* in proper sequence. The first of these conditions concerns the requirement that the proper recursive forms of the CGI-equations for propagating the solution in the different directions, do propagate *strictly* in these directions. For example, for propagation to the left, the coordinates of the left-extremal terms in the CGI-equation must lie strictly to the left of *all* other terms in that equation. A similar condition must hold for propagation in the other directions. A second condition requires that the core should be *sufficiently large*: the terms in the CGI-equation whose extremal term is located at the outer boundary of the core, must all fall inside the core. Once a core is found, one has to solve the set of seed-internal equations: This also determines the freecells and depcells in the seed. Working things out however, can be more cumbersome than in the one-dimensional CGI-case. The following is a generic example.

Example 3. $q^n = 2^1$; $2 \times 2 \times 2$ cubic tile. CA-rule: $R(X, Y) = X + Y + XY$. $w(0, 0, 0) = w(0, 1, 0) = w(1, 0, 1) = w(1, 1, 1) = 1$, all other $w(r, i, j)$-values are 0. The neighborhood is $\mathcal{N}_k = \{(-1, 0), (0, 0), (-1, 1), (0, 1)\}$ [the one shown in Fig. 10(k)].

The corresponding CGI-equations are:

$\forall(k,\, l) \in \mathbb{Z}^2$:

$$\begin{aligned} a(k-1,\, l) + a(k,\, l) + a(k-1,\, l+1) + a(k,\, l+1) \\ = S\Big[a\Big(\frac{k}{2} + a_{00},\, \frac{l}{2} + b_{00}\Big)\Big| \\ a\Big(\frac{k-1}{2} + a_{10},\, \frac{l}{2} + b_{10}\Big)\Big| \\ a\Big(\frac{k}{2} + a_{01},\, \frac{l-1}{2} + b_{01}\Big)\Big| \\ a\Big(\frac{k-1}{2} + a_{11},\, \frac{l-1}{2} + b_{11}\Big)\Big], \end{aligned} \tag{32}$$

where $S[a|b|c|\cdots]$ indicates that the proper term should be selected in dependence on the parity of the variables k and l (for example, when $k =$ even, $l =$ odd, the third element in $S[\cdots]$ must be taken).

We will opt for a *unidirectional core* which will contain the blue core from Fig. 10(k). In that case, the propagation of the solution occurs along a computational front in the left (-upward) and right(-downward) directions.

It will be appropriate to make a *coordinate change*, by introducing a coordinate axis which is orthogonal to the core direction. So, let's change the $(k,\, l)$-coordinates to $(k,\, \lambda = k+l)$. Then $a(k,\, l)$ in the original coordinate system will be written as $a_\lambda(k) = a_{k+l}(k)$ in the new $(k,\, \lambda)$-coordinate system [see Fig. 14]. With

$$c_{\sigma_1,\sigma_2} \stackrel{\text{def}}{=} a_{\sigma_1,\sigma_2} + b_{\sigma_1,\sigma_2}\,,$$

the CGI-equations displayed in Eq. (35) transform into

$\forall(k,\, l) \in \mathbb{Z}^2$:

$$\begin{aligned} a_{\lambda-1}(k-1) + a_\lambda(k) + a_\lambda(k-1) + a_{\lambda+1}(k) \\ = S\Big[a_{\frac{\lambda}{2}+c_{00}}\Big(\frac{k}{2} + a_{00}\Big)\Big|a_{\frac{\lambda-1}{2}+c_{10}}\Big(\frac{k-1}{2} + a_{10}\Big)\Big| \\ a_{\frac{\lambda-1}{2}+c_{01}}\Big(\frac{k}{2} + a_{01}\Big)\Big|a_{\frac{\lambda-2}{2}+c_{11}}\Big(\frac{k-1}{2} + a_{11}\Big)\Big]. \end{aligned} \tag{33}$$

We will try to solve this set of equations recursively in the variable λ by considering a seed of the form $\{a_\lambda(k) | k \in \mathbb{Z},\, \lambda \in I[\underline{\lambda}+1,\, \overline{\lambda}-1]\}$, where $\overline{\lambda} - \underline{\lambda}$ is minimal (see [Barbé, 1996a]). This seed will then be expanded layer by layer to the left starting with $a_{\underline{\lambda}}(\cdot)$ and to the right starting with $a_{\overline{\lambda}}(\cdot)$, using Eq. (33) in a *strictly* left- and right-propagating form.

The leftmost extremal neighborhood term in Eq. (33) is $a_{\lambda-1}(k-1)$. As $(k,\, \lambda)$ are dummies, let us renotate this term as $a_\lambda(k)$, then the *left-propagation form* of Eq. (33) is given by

$$\begin{aligned} a_\lambda(k) = a_{\lambda+1}(k+1) + a_{\lambda+1}(k) + a_{\lambda+2}(k+1) \\ + S\Big[a_{\frac{\lambda+1}{2}+c_{00}}\Big(\frac{k+1}{2} + a_{00}\Big)\Big|a_{\frac{\lambda}{2}+c_{10}}\Big(\frac{k}{2} + a_{10}\Big)\Big| \\ a_{\frac{\lambda}{2}+c_{01}}\Big(\frac{k+1}{2} + a_{01}\Big)\Big|a_{\frac{\lambda-1}{2}+c_{11}}\Big(\frac{k}{2} + a_{11}\Big)\Big]. \end{aligned} \tag{34}$$

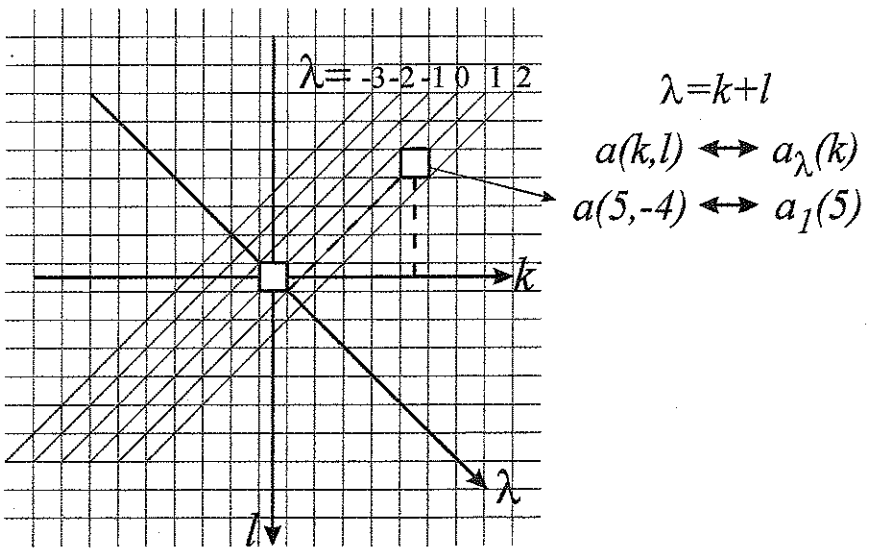


Fig. 14. The new coordinate system for Example 3. $(k,\, l)$: Old coordinates, $(k,\, \lambda)$: New coordinates.

(Recall that these are all equations in $\mathbb{Z}_2$, so all arithmetic is mod 2.)

These equations are *strictly* left-propagating only if all λ-related indices in its right side are greater than λ, i.e. when

$$\lambda < \min\left(\frac{\lambda+1}{2}+c_{00}, \frac{\lambda}{2}+c_{10}, \frac{\lambda}{2}+c_{01}, \frac{\lambda-1}{2}+c_{11}\right),$$

or, when

$$\lambda < \min(2c_{00}+1,\, 2c_{10},\, 2c_{01},\, 2c_{11}-1)\,. \tag{35}$$

The maximal value of λ as allowed by Eq. (35) is then also the maximal value of $\underline{\lambda}$, the left outer boundary coordinate of the seed:

$$\underline{\lambda} \le \min(2c_{00}+1,\, 2c_{10},\, 2c_{01},\, 2c_{11}-1) - 1\,. \tag{36}$$

The right extremal term in Eq. (33) is $a_{\lambda+1}(k)$. Replacing it by $a_\lambda(k)$, we find the *right-propagating form* of this equation:

$$\begin{aligned} a_\lambda(k) = {} & a_{\lambda-2}(k-1)+a_{\lambda-1}(k)+a_{\lambda-1}(k-1) \\ & + S\left[a_{\frac{\lambda-1}{2}+c_{00}}\left(\frac{k}{2}+a_{00}\right)\middle| a_{\frac{\lambda-2}{2}+c_{10}}\left(\frac{k-1}{2}+a_{10}\right)\middle| \right. \\ & \left. a_{\frac{\lambda-2}{2}+c_{01}}\left(\frac{k}{2}+a_{01}\right)\middle| a_{\frac{\lambda-3}{2}+c_{11}}\left(\frac{k-1}{2}+a_{11}\right)\right]. \end{aligned} \tag{37}$$

Strict right-propagation occurs only if

$$\lambda > \max\left(\frac{\lambda-1}{2}+c_{00}, \frac{\lambda-2}{2}+c_{10}, \frac{\lambda-2}{2}+c_{01}, \frac{\lambda-3}{2}+c_{11}\right),$$

which leads in its turn to the following lower bound for the right outer boundary coordinate of the seed:

$$\overline{\lambda} \ge \max(2c_{00}-1,\, 2c_{10}-2,\, 2c_{01}-2,\, 2c_{11}-3)+1\,. \tag{38}$$

But there are still other conditions on $\underline{\lambda}$ and $\overline{\lambda}$: The seed should be wide enough in the following sense. $a_{\underline{\lambda}}(k)$ can only be the first column to be determined from the seed using the left-propagation form of Eq. (34), when all terms in this equation are all in the seed, i.e. when the maximal value of the λ-related coordinates does not exceed $\overline{\lambda}-1$. This implies that

$$\max\left(\underline{\lambda}+2, \frac{\underline{\lambda}}{2}+c_{10}, \frac{\underline{\lambda}}{2}+c_{01}\right) \le \overline{\lambda}-1 \quad \text{when} \quad \underline{\lambda} = \text{even} \tag{39a}$$

$$\max\left(\underline{\lambda}+2, \frac{\underline{\lambda}+1}{2}+c_{00}, \frac{\underline{\lambda}-1}{2}+c_{11}\right) \le \overline{\lambda}-1 \quad \text{when} \quad \underline{\lambda} = \text{odd} \tag{39b}$$

A similar symmetric condition holds for the right-propagating Eq. (37):

$$\min\left(\overline{\lambda}-2, \frac{\overline{\lambda}-2}{2}+c_{10}, \frac{\overline{\lambda}-2}{2}+c_{01}\right) \le \underline{\lambda}+1 \quad \text{when} \quad \overline{\lambda} = \text{even} \tag{40a}$$

$$\min\left(\overline{\lambda}-2, \frac{\overline{\lambda}-1}{2}+c_{00}, \frac{\overline{\lambda}-3}{2}+c_{11}\right) \le \underline{\lambda}+1 \quad \text{when} \quad \overline{\lambda} = \text{odd} \tag{40b}$$

Let us now fix some values for the parameters a_{σ_1,σ_2} and b_{σ_1,σ_2}:

$$\begin{aligned} a_{00} = b_{00} = -1\,, \\ a_{10} = b_{10} = a_{01} = b_{01} = a_{11} = b_{11} = 1\,. \end{aligned} \tag{41}$$

This makes $c_{00} = -2$, $c_{01} = c_{10} = c_{11} = 2$.

Explicitizing the conditions of Eqs. (36)–(40) leads to

$$\underline{\lambda} \le -4 \quad \text{and} \quad \overline{\lambda} \ge 3\,,$$

and so the minimal seed is

$$\mathcal{S} = \{a_\lambda(k) | k \in \mathbb{Z},\, \lambda = -3, -2, -1, 0, 1, 2\}\,.$$

In order to find the depcells in this seed, we need to look for the *seed-internal equations* (i.e. the set of all CGI-equations that have all terms within the seed). In left-propagating form, these seed-internal equations are

$$\begin{aligned} a_{-3}(k) = {} & a_{-2}(k+1)+a_{-2}(k)+a_{-1}(k+1) \\ & + S\left[a_{-3}\left(\frac{k-1}{2}\right)\middle| a_0\left(\frac{k+2}{2}\right)\right] \end{aligned} \tag{42}$$

$$\begin{aligned} a_{-2}(k) = {} & a_{-1}(k+1)+a_{-1}(k)+a_0(k+1) \\ & + S\left[a_1\left(\frac{k+2}{2}\right)\middle| a_1\left(\frac{k+3}{2}\right)\right] \end{aligned} \tag{43}$$

$$\begin{aligned} a_{-1}(k) = {} & a_0(k+1)+a_0(k)+a_1(k+1) \\ & + S\left[a_{-2}\left(\frac{k-1}{2}\right)\middle| a_1\left(\frac{k+2}{2}\right)\right] \end{aligned} \tag{44}$$

$$\begin{aligned} a_0(k) = {} & a_1(k+1)+a_1(k)+a_2(k+1) \\ & + S\left[a_2\left(\frac{k+2}{2}\right)\middle| a_2\left(\frac{k+3}{2}\right)\right] \end{aligned} \tag{45}$$

These equations can be solved as follows:

- let $\{a_1(k), a_2(k) | k \in \mathbb{Z}\}$ be freecells initially.

- $a_0(\cdot)$ can be calculated in a direct manner from $a_1(k)$ and $a_2(k)$ using Eq. (45).
- Equations (43) and (44) show that $a_{-1}(\cdot)$ and $a_{-2}(\cdot)$ are *jointly recursive* in a_{-1} and a_{-2} [with a rescaling term in a_{-2}], while the other terms in $a_0(\cdot)$ and $a_1(\cdot)$ are already known from the previous steps. This pair of equations can be solved recursively (in k) to the left (i.e. for decreasing k) from a certain value $\underline{k}$ on, according to the sequence

$$a_{-1}(\underline{k}),\, a_{-2}(\underline{k}),\, a_{-1}(\underline{k}-1),\, a_{-2}(\underline{k}-1), \ldots \tag{46}$$

when the following strict left-propagating condition for $a_{-1}(k)$ in Eq. (44) holds: $k < \frac{k-1}{2}$, or equivalently $k < -1$, when k is odd.
This implies that

$$\underline{k} \leq -2\,. \tag{47}$$

The equations under consideration can also be right propagated according to the sequence

$$a_{-1}(\overline{k}+1),\, a_{-2}(\overline{k}),\, a_{-1}(\overline{k}+2),\, a_{-2}(\overline{k}+1), \ldots \tag{48}$$

as Eq. (43) shows that $a_{-1}(k+1)$ [and so $a_{-1}(k)$] has to be determined before $a_{-2}(k)$. $\overline{k}$ is determined by the strict right-propagating conditions for $a_{-1}(k+1)$ in Eq. (44): $k+1 > \frac{(k+1)-1}{2}$, or equivalently $k > -2$, for k even; as $a_{-1}(k+1)$ must precede $a_{-2}(k)$, it must also hold that $k > \frac{(k+1)-1}{2}$, or $k > 0$. So,

$$\overline{k} \geq 1\,. \tag{49}$$

The seed for solving Eqs. (43) and (44) consists of the cells which are *not* in the largest possible sets given by Eqs. (46)–(49), i.e.

$$\mathcal{S}_{-1,-2} = \{a_{-1}(-1),\, a_{-2}(-1),\, a_{-1}(0),\, a_{-2}(0),\, a_{-1}(1)\}\,.$$

Inspection shows readily that the seed is sufficiently large to start the left- and right propagation of Eqs. (43) and (44). Again, there may be depcells in this seed. In order to find these, we consider the seed-internal equations in $a_{-1}(\cdot)$ and $a_{-2}(\cdot)$. These are [from Eqs. (43) and (44)]:

$$a_{-1}(-1) = a_0(0) + a_0(-1) + a_1(0) + a_{-2}(-1) \tag{50a}$$

$$a_{-2}(-1) = a_{-1}(0) + a_{-1}(-1) + a_0(0) + a_1(1) \tag{50b}$$

$$a_{-1}(0) = a_0(1) + a_0(0) + a_1(1) + a_1(1) \tag{50c}$$

$$a_{-2}(0) = a_{-1}(1) + a_{-1}(0) + a_0(1) + a_1(1) \tag{50d}$$

$$a_{-1}(1) = a_0(2) + a_0(1) + a_1(2) + a_{-2}(0) \tag{50e}$$

The solution ($\mathbb{Z}_2$-arithmetic!), in terms of the already known $a_0(\cdot)$ and $a_1(\cdot)$ is found to be:

$$a_{-2}(-1) = \text{free} \tag{51a}$$

$$a_{-1}(-1) = a_0(0) + a_0(-1) + a_1(0) + a_{-2}(-1) \tag{51b}$$

$$a_{-1}(0) = a_0(1) + a_0(0) \tag{51c}$$

$$a_{-2}(0) = \text{free} \tag{51d}$$

$$a_{-1}(1) = a_0(2) + a_0(1) + a_1(2) + a_{-2}(0) \tag{51e}$$

That $a_{-2}(-1)$ can be chosen freely follows from the addition of Eqs. (50a)–(50c), yielding

$$a_{-2}(-1) = a_0(-1) + a_0(0) + a_0(1) + a_1(0) + a_1(1) + a_{-2}(-1)\,,$$

which produces the condition

$$a_0(-1) + a_0(0) + a_0(1) + a_1(0) + a_1(1) = 0\,. \tag{52}$$

The freeness of $a_{-2}(0)$ follows from adding Eqs. (50c)–(50e) and leads to the condition

$$a_0(0) + a_0(1) + a_0(2) + a_1(1) + a_1(2) = 0\,. \tag{53}$$

Writing the $a_0(\cdot)$-terms in Eqs. (52) and (53) as a function of $a_1(\cdot)$ and $a_2(\cdot)$ using Eq. (45) gives respectively:

$$a_2(1) = a_2(0) + a_1(-1) + a_1(0) + a_1(1) + a_1(2) \tag{54}$$

and

$$a_2(3) = a_1(0) + a_1(1) + a_1(2) + a_1(3) + a_2(2)\,. \tag{55}$$

These conditions turn $a_2(1)$ and $a_2(3)$ into depcells.

- Now remains $a_{-3}(k)$ to be determined, using Eq. (42). This equation is autorecursive in $a_{-3}(\cdot)$ with a rescaling term. It can be solved through left- and right-propagation starting with the seedcell-state $a_{-3}(-1)$, which can be selected freely. As $a_{-3}(-1)$ has to satisfy Eq. (42), it follows that

$$a_{-3}(-1) = a_{-2}(0) + a_{-2}(-1) + a_{-1}(0) + a_{-3}(-1)\,,$$

which creates the additional condition

$$a_{-2}(0) = a_{-2}(-1) + a_{-1}(0) \overset{51(c)}{=} a_{-2}(-1) + a_0(1) + a_0(0)\,. \quad (56)$$

This makes a depcell from $a_{-2}(0)$ [it was a freecell before, see Eq. (51d)].

In summary, we have found that the CGI-equations [Eq. (32)] for the specific parameters given by Eq. (41), can be solved recursively by using the left- and right-propagating forms of these equations [Eq. (34) and Eq. (37)], starting from the seed

$$\mathcal{S} = \{a_\lambda(k) | k \in \mathbb{Z},\ \lambda = -3,\ -2,\ -1,\ 0,\ 1,\ 2\}$$

whose freecell-set is

$$\mathcal{F} = \{a_1(k) | k \in \mathbb{Z}\} \cup \{a_2(k) | k \in \mathbb{Z}\backslash\{1,\ 3\}\} \cup \{a_{-2}(-1),\ a_{-3}(-1)\}\,.$$

The depcells form the set $\mathcal{D} = \mathcal{S} \setminus \mathcal{F}$, and can be determined using Eqs. (54), (55), (45), (51b), (51c), (51e), (56), (44), (43) and (42), in that order.

Figure 15(a) displays a part of this solution for the case where all freecell-states are put to 1 (constant-freecell case). The solution for randomly attributed freecell-states is very similar to the one in Fig. 15(b) (which is actually the solution for a random-freecell *bidirectional* seed, as derived in the next example).

Thus far the example featuring a *unidirectional* core. It is generic in the sense that a seed for solving the CGI-equations can always be found by using the conditions of strict propagation in opposite directions, together with the condition of sufficient seed-width. These conditions can be applied separately for the two coordinates in proper sequence. The same principles also work when a *bidirectional* core is chosen: There is always one direction in the core which is a main direction in the sense that the solution develops along computational fronts which move in the complementary direction. The seed-cells along this complementary direction merely serve as initiatiors for propagating

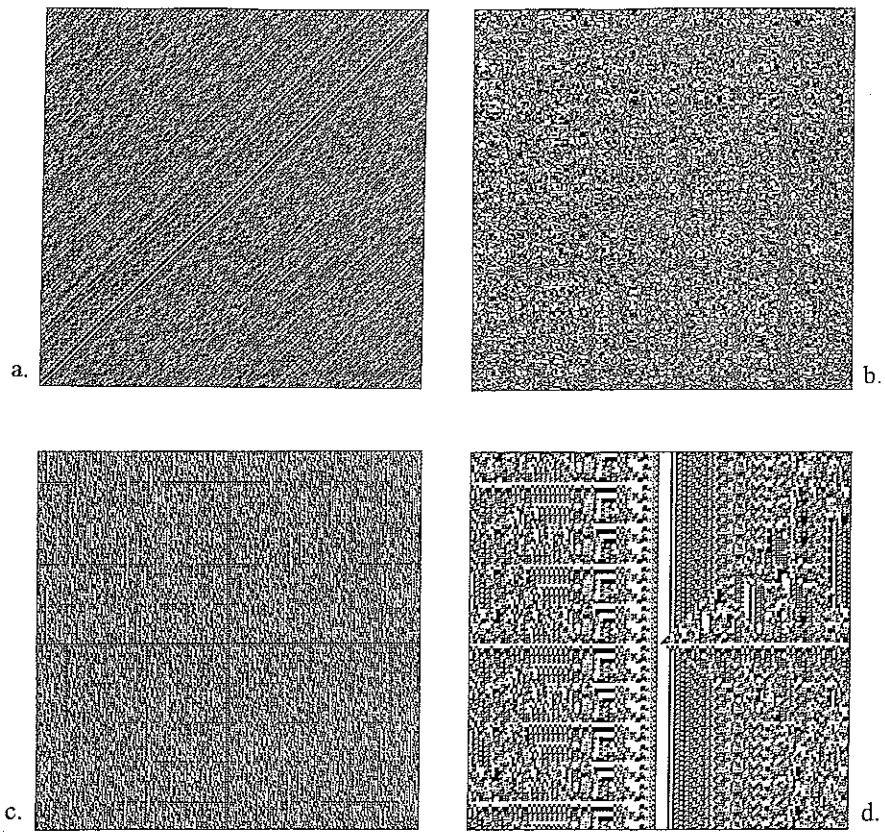


Fig. 15. Different initial configuration solutions for the CGI-problem of Examples 3 and 4: (a) with a unidirectional seed, all freecells are in state 1 (Example 3); (b) with a bidirectional seed, random freecells (Example 4); (c) with a bidirectional seed, all freecells are in state 1 (Example 4). (d) with a bidirectional seed, all freecells are in state 0, except $a(-2,\ 0)$ and $a(2,\ 1)$ which are in state 1.

the solution along the computational front in parallel to the main direction. This is illustrated in the following example.

Example 4. It concerns the same CGI-problem as in Example 3, but now the solution will be generated by a *bidirectional* seed. As a core for the corresponding equation without rescaling term has two orthogonal single core-lines [see the red lines in Fig. 10(k)], we propose a seed of the form

$$\{a(k,\, l)|l \in \mathbb{Z},\, k \in I[\underline{k}+1,\, \overline{k}-1]\} \cup \{a(k,\, l_0)|k \in \mathbb{Z}\}\,, \tag{57}$$

where $\underline{k}$, $\overline{k}$, l_0 are to be determined further.

Starting with this seed, the solution will be propagated as follows:

- propagation to the left: Calculate $a(k, \cdot)$ for $k = \underline{k},\, \underline{k}-1,\, \underline{k}-2, \ldots$. For a given k, $a(k,\, l)$ will be propagated for decreasing l (upward direction) first, according to the sequence

  $$a(k,\, l_0-1),\, a(k,\, l_0-2),\, a(k, l_0-3), \ldots$$

  and then for increasing l (downward direction):

  $$a(k,\, l_0+1),\, a(k,\, l_0+2),\, a(k,\, l_0+3), \ldots$$

- propagation to the right occurs in a symmetric way: Calculate $a(k, \cdot)$ for $k = \overline{k},\, \overline{k}+1,\, \overline{k}+2, \ldots$. For a given k, propagate $a(k,\, l)$ upwards and downwards starting from $a(k,\, l_0)$.

The *left-upward* form of the CGI-equation is given by

$$\begin{aligned} a(k,\, l) = {} & a(k,\, l+1) + a(k+1,\, l) + a(k+1,\, l+1) \\ & + S\left[a\left(\frac{k-1}{2},\, \frac{l-2}{2}\right)\middle|a\left(\frac{k+2}{2},\, \frac{l+2}{2}\right)\middle| \right. \\ & \left. a\left(\frac{k+3}{2},\, \frac{l+1}{2}\right)\middle|a\left(\frac{k+2}{2},\, \frac{l+1}{2}\right)\right]. \end{aligned} \tag{58}$$

We also need

- the *left-downward* propagation form

$$\begin{aligned} a(k,\, l) = {} & a(k,\, l-1) + a(k+1,\, l) + a(k+1,\, l-1) \\ & + S\left[a\left(\frac{k-1}{2},\, \frac{l-3}{2}\right)\middle|a\left(\frac{k+2}{2},\, \frac{l+1}{2}\right)\middle| \right. \\ & \left. a\left(\frac{k+3}{2},\, \frac{l}{2}\right)\middle|a\left(\frac{k+2}{2},\, \frac{l}{2}\right)\right]. \end{aligned} \tag{59}$$

- the *right-upward* propagation form

$$\begin{aligned} a(k,\, l) = {} & a(k,\, l+1) + a(k-1,\, l) + a(k-1,\, l+1) \\ & + S\left[a\left(\frac{k-2}{2},\, \frac{l-2}{2}\right)\middle|a\left(\frac{k+1}{2},\, \frac{l+2}{2}\right)\middle| \right. \\ & \left. a\left(\frac{k+2}{2},\, \frac{l+1}{2}\right)\middle|a\left(\frac{k+1}{2},\, \frac{l+1}{2}\right)\right]. \end{aligned} \tag{60}$$

- the *right-downward* propagation form

$$\begin{aligned} a(k,\, l) = {} & a(k,\, l-1) + a(k-1,\, l) + a(k-1,\, l-1) \\ & + S\left[a\left(\frac{k-2}{2},\, \frac{l-3}{2}\right)\middle|a\left(\frac{k+1}{2},\, \frac{l+1}{2}\right)\middle| \right. \\ & \left. a\left(\frac{k+2}{2},\, \frac{l}{2}\right)\middle|a\left(\frac{k+1}{2},\, \frac{l}{2}\right)\right]. \end{aligned} \tag{61}$$

Strict left-propagation (decreasing k) of Eqs. (58) and (59) requires that $a(k,\, l+1)$ [respectively $a(k,\, l-1)$] is known and that $k < \min(\frac{k-1}{2},\, \frac{k+2}{2},\, \frac{k+3}{2})$, implying that $\underline{k} \leq -2$. Similarly, strict right-propagation of Eqs. (60) and (61) imposes the condition that $a(k,\, l+1)$ [respectively $a(k,\, l-1)$] be known and that $k > \max(\frac{k-2}{2},\, \frac{k+1}{2},\, \frac{k+2}{2})$, implying that $\overline{k} \geq 3$. So, the vertical part of the seed proposed in Eq. (57) is

$$\{a(k,\, l)|l \in \mathbb{Z},\, k = -1,\, 0,\, 1,\, 2\}\,.$$

(The fact that $a(k,\, l+1)$ and $a(k,\, l-1)$ are known is guaranteed by fixing the horizontal part of the seed, i.e. $\{a(k,\, l_0)|k \in \mathbb{Z}\}$.)

It is readily checked that this vertical part of the seed is wide enough (if it were not, this requirement would widen the seed). Are there seed-internal equations which may possibly create dep-cells? Yes, they are [in right-upward propagation form, from Eq. (60)]:

$$\begin{aligned} a(0,\, l) = {} & a(0,\, l+1) + a(-1,\, l) + a(-1,\, l+1) \\ & + S\left[a\left(-1,\, \frac{l-2}{2}\right)\middle|a\left(1,\, \frac{l+1}{2}\right)\right] \end{aligned} \tag{62a}$$

$$\begin{aligned} a(1,\, l) = {} & a(1,\, l+1) + a(0,\, l) + a(0,\, l+1) \\ & + S\left[a\left(1,\, \frac{l+2}{2}\right)\middle|a\left(1,\, \frac{l+1}{2}\right)\right] \end{aligned} \tag{62b}$$

$$\begin{aligned} a(2,\, l) = {} & a(2,\, l+1) + a(1,\, l) + a(1,\, l+1) \\ & + S\left[a\left(0,\, \frac{l-2}{2}\right)\middle|a\left(2,\, \frac{l+1}{2}\right)\right] \end{aligned} \tag{62c}$$

The right-downward propagation form of these seed-internal equations is [from Eq. (61)]:

$$a(0, l) = a(0, l-1) + a(-1, l) + a(-1, l-1) + S\left[a\left(-1, \frac{l-3}{2}\right)\middle| a\left(1, \frac{l}{2}\right)\right] \quad (63a)$$

$$a(1, l) = a(1, l-1) + a(0, l) + a(0, l-1) + S\left[a\left(1, \frac{l+1}{2}\right)\middle| a\left(1, \frac{l}{2}\right)\right] \quad (63b)$$

$$a(2, l) = a(2, l-1) + a(1, l) + a(1, l-1) + S\left[a\left(0, \frac{l-3}{2}\right)\middle| a\left(2, \frac{l}{2}\right)\right] \quad (63c)$$

Let us choose $\{a(-1, l)|l \in \mathbb{Z}\}$ as initial freecells. Then these seed-internal equations can be solved recursively (in l). Observe that $a(0, \cdot)$ and $a(1, \cdot)$ are jointly recursive with rescaling terms in $a(1, \cdot)$. These can be solved first from a seed $\{a(0, l), a(1, l)|l \in I[\underline{l}+1, \bar{l}-1]\}$ according to the upward propagating sequence [using Eqs. (62a) and (62b)]

$$a(0, \underline{l}),\ a(1, \underline{l}),\ a(0, \underline{l}-1),\ a(1, \underline{l}-1),\ a(0, \underline{l}-2),\ a(1, \underline{l}-3), \ldots$$

and according to the downward propagating sequence [using Eqs. (63a) and (63b)]

$$a(0, \bar{l}),\ a(1, \bar{l}),\ a(0, \bar{l}+1),\ a(1, \bar{l}+1), \ldots .$$

$\underline{l}$ and $\bar{l}$ follow from the strict upward and downward propagation conditions in $a(0, \cdot)$ and $a(1, \cdot)$ for Eqs. (62) and (63):

$$l < \min\left(\frac{l+1}{2}, \frac{l+2}{2}\right), \quad \text{implying} \quad \underline{l} \leq 0,$$

and

$$l > \max\left(\frac{l}{2}, \frac{l+1}{2}\right), \quad \text{implying} \quad \bar{l} \geq 2.$$

So, the minimal seed for solving $a(0, l)$ and $a(1, l)$, which is also found to be sufficiently wide by inspection, is $\{a(0, 1), a(1, 1)\}$. Further inspection shows that there are no seed-internal equations, and so $a(0, 1)$ and $a(1, 1)$ can be chosen freely.

$a(2, l)$ can be solved in a recursive way using Eqs. (62c)–(63c), starting from the seed $a(2, 1)$.

Summarizing, a bidirectional seed for the CGI-problem under consideration is given by

$$\mathcal{S} = \{a(-1, l), a(0, l), a(1, l), a(2, l)|l \in \mathbb{Z}\} \cup \{a(k, 1)|k \in \mathbb{Z}\}.$$

The freecell set is

$$\mathcal{F} = \{a(-1, l)|l \in \mathbb{Z}\} \cup \{a(k, 1)|k \in \mathbb{Z}\}.$$

The other cells in $\mathcal{S}\backslash\mathcal{F}$ are depcells, and are to be determined using Eqs. (62) and (63). From this seed, the whole solution $(a(k, l))_{(k,l)\in\mathbb{Z}^2}$ can be propagated using Eqs. (58)–(61).

Figure 15(b) shows a part of the solution using the core derived in this example, whereby the freecells in the seed have randomly attributed states. Figure 15(c) shows the solution for which the freecells all have the value 1. Figure 15(d) still shows another solution.

Thus far the example concerning a bidirectional seed for solving the CGI-equations under consideration.

Additional remarks concerning Examples 3 and 4

Observe that the set of freecells in Examples 3 and 4 could have been selected in a different way (more in particular, the set of cells that have been chosen as freecells at the initial stage of solving the seed-internal equations could have been selected in another way).

When one is interested in finding solutions which are rather "structured", like in Figs. 15(a) and 15(c), where for example the freecells along certain core-lines are periodic sequences, it is interesting to consider more than one type of core (and associated seed): it would have been difficult to obtain the solution of Fig. 15(c) from a unidirectional seed, unless through an exhaustive exploration of all possible unidirectional seed-configurations. In the same sense, the solution displayed in Fig. 15(a) would not have been easily found from a bidirectional seed.

The past two examples show the way for finding (two-dimensional) CGI-solutions in all other cases. However, working things out may generally be much more involved: The number of jointly recursive seed-internal equations and the order in which seed-internal equations are to be solved may vary largely from case to case in dependence on the problem parameters. Developing a computer program for solving two-dimensional CGI-problems in general seems feasible, but it will be more difficult than for the one-dimensional case. This makes it hard for the moment to explore the space of possible two-dimensional CGI-solutions in the same way as was possible for the one-dimensional case (see [Barbé,

1996a]). Nevertheless, we will discuss in Sec. 5 certain morphological aspects of some CGI-orbits from a few dozen CGI-problems that were solved.

4.3. *Finite seed restrictions: Heredity of IGIP-C and IGIP-nC class membership*

Consider a finitely restricted seed for the full CGI-equations in the same sense as was done for the equations without rescaling term. As non-collinearity of the renormalizing neighborhood implies that the restricted seed solution for the CGI-equations without rescaling term is finite too, this will remain so in case the rescaling term is taken into account. So, if the CGI-equation without rescaling term belongs to the IGIP-nC class, the same is true for the full CGI-equation.

Will IGIP-C class membership be preserved for CGI-equations with a collinear neighborhood when we move from the equations without rescaling term towards the full equations? i.e. will a finitely restricted seed also produce a solution with infinite support? In spite of the fact that the rescaling term is *not* collinear with the renormalizing neighborhood, the answer is affirmative, at least with certain conditions on the seed-restriction (such conditions were not necessary in the case "without rescaling term"). This will be stated as a next theorem. We first need to establish the concept of a *seed-kernel.*

Seed-kernel, kernel-containing seed restriction

Consider the CGI-equations with a collinear renormalizing neighborhood that runs parallel to the k-axis (and restricted to the case $\mathbb{Z}_2$):

$\forall (k, l) \in \mathbb{Z}^2$:

$$\sum_{j\in\mathcal{J}\subset\mathbb{Z}} a(k+j, l) = S\Big[a\Big(\frac{k}{2}+a_{00}, \frac{l}{2}+b_{00}\Big)\Big| a\Big(\frac{k-1}{2}+a_{10}, \frac{l}{2}+b_{10}\Big)\Big| a\Big(\frac{k}{2}+a_{01}, \frac{l-1}{2}+b_{01}\Big)\Big| a\Big(\frac{k-1}{2}+a_{11}, \frac{l-1}{2}+b_{11}\Big)\Big]. \tag{64}$$

The left and right propagating forms (in k) of these equations are:

- *left propagating form*:

$$a(k, l) = \sum_{j\in\mathcal{J}\setminus\{j_{\min}\}} a(k-j_{\min}+j, l) + S\Big[a\Big(\frac{k+\mu_{\sigma_1,\sigma_2}}{2}, \frac{l+\nu_{\sigma_1,\sigma_2}}{2}\Big)\Big]_{\sigma_1,\sigma_2}, \tag{65}$$

where

$$j_{\min} = \text{minimal index in } \mathcal{J}$$
$$\mu_{\sigma_1,\sigma_2} = j_{\min} - \sigma_1 + 2a_{\sigma_1,\sigma_2}$$
$$\nu_{\sigma_1,\sigma_2} = -\sigma_2 + 2b_{\sigma_1,\sigma_2}$$

and $S[a(.,.)]_{\sigma_1,\sigma_2}$ means that the proper selection of σ_1, σ_2 should be taken.

- *right propagating form*:

$$a(k, l) = \sum_{j\in\mathcal{J}\setminus\{j_{\max}\}} a(k-j_{\max}+j, l) + S\Big[a\Big(\frac{k+\mu'_{\sigma_1,\sigma_2}}{2}, \frac{l+\nu_{\sigma_1,\sigma_2}}{2}\Big)\Big]_{\sigma_1,\sigma_2}, \tag{66}$$

where

$$j_{\max} = \text{maximal index in } \mathcal{J}$$
$$\mu'_{\sigma_1,\sigma_2} = j_{\max} - \sigma_1 + 2a_{\sigma_1,\sigma_2}.$$

Let

$$\{a(k, l)|k \in I[\underline{k}+1, \overline{k}-1], l \in \mathbb{Z}\}$$

be a seed from which $(a(k, l))_{(k,l)\in\mathbb{Z}^2}$ can be solved, using Eqs. (65) and (66). Recall from the Examples 3 and 4 that $\overline{k}$ and $\underline{k}$ are integers such that $\overline{k}-\underline{k}$ is minimal and satisfy the conditions for *strict* left and right propagation for these equations, and also the condition of sufficient seed-width:

$$\underline{k} \le \min_{\sigma_1,\sigma_2}(\mu_{\sigma_1,\sigma_2}) - 1 \tag{67a}$$
$$\overline{k} \ge \max_{\sigma_1,\sigma_2}(\mu'_{\sigma_1,\sigma_2}) + 1 \tag{67b}$$
$$\underline{k} - j_{\min} + j_{\max} \le \overline{k} - 1 \tag{67c}$$
$$\overline{k} - j_{\min} + j_{\max} \ge \underline{k} + 1. \tag{67d}$$

Let

$$\underline{l} = \min_{\sigma_1,\sigma_2}(\nu_{\sigma_1,\sigma_2}) \quad \text{and} \quad \overline{l} = \max_{\sigma_1,\sigma_2}(\nu_{\sigma_1,\sigma_2}). \tag{68}$$

Definition 3. The *seed-kernel* is that part of the seed restricted to

$$\mathcal{S}_{\text{kernel}} = \{a(k, l)|k \in I[\underline{k}+1, \overline{k}-1], l \in I[\underline{l}, \overline{l}]\}. \tag{69}$$

A *kernel-containing seed restriction* $\mathcal{S}'$ is then defined as any part of the seed which contains $\mathcal{S}_{\text{kernel}}$. It is of the form

$$\mathcal{S}' = \{a(k, l) | k \in I[\underline{k}+1, \overline{k}-1], l \in I[l_{\min}, l_{\max}]\}, \tag{70}$$

where

$$l_{\min} \leq \underline{l} \quad \text{and} \quad l_{\max} \geq \overline{l}. \tag{71}$$

Theorem 4. *Given the CGI-equations with collinear renormalizing neighborhood of the form given in Eq. (64), a kernel-containing seed restriction as given by Eq. (70) generates the solution* $(a(k, l))_{(k,l)\in\mathbb{Z}\times I[l_{\min},l_{\max}]}$.

Proof. Consider the left-propagating form of the CGI-equation [Eq. (65)] for $a(k, l)$ in the set

$$\mathcal{L}_{\underline{k}} = \{a(\underline{k}, l) | l_{\min} \leq l \leq l_{\max}\}$$

(this is the left outer boundary layer of the kernel-containing seed $\mathcal{S}'$). The conditions in Eqs. (67), (68) and (71) guarantee that the corresponding rescaling term for each of the elements in this set lies inside $\mathcal{S}'$, and so do the complementary neighborhood terms [these are the terms gathered in the $\sum$-symbol in Eq. (65)]. So all elements in $\mathcal{L}_{\underline{k}}$ can be calculated, and this extends the restricted seed one layer to the left. It is not possible to calculate $a(\underline{k}, l)$ for $l > l_{\max}$ or $l < l_{\min}$, because the complementary neighborhood terms are then no longer in $\mathcal{S}'$.

Because of the same conditions [Eqs. (67), (68) and (71)], the known solution $\mathcal{S}' \cup \mathcal{L}_{\underline{k}}$ can be expanded to the left with another layer $\mathcal{L}_{\underline{k}-1} = \{a(\underline{k}-1, l) | l_{\min} \leq l \leq l_{\max}\}$, then with the layer $\mathcal{L}_{\underline{k}-2}$, $\mathcal{L}_{\underline{k}-3}$, etc. So, the kernel-containing seed can be expanded to the left in an unlimited way.

Similar arguments for the right-propagating form allow unbounded growth of the solution to the right. ■

A similar theorem can be formulated when the renormalizing neighborhood terms are aligned along the direction of the line with equation $l = \frac{b}{a}k$ (a, b integers), but in a modified orthogonal coordinate system (κ, λ), in which $\kappa = al - bk$ and $\lambda = bl + ak$ (i.e. one coordinate axis in parallel with the corelines and the other orthogonal to it). The proof proceeds along the same lines. However, the bookkeeping of the (κ, λ)-related equations and conditions is more cumbersome.

5. Some Morphology: Complex Order from Complete Disorder and from Simple Order

It is practically impossible to give a complete idea of the diversity and complexity of orbit patterns that are coarse-graining invariant. Dependent on the tile-configuration parameters Ξ and on the shift-parameters a_{σ_1,σ_2}, b_{σ_1,σ_2}, the patterns may show structures of diverse complexity, some of them rather unstructured, others exhibiting a complexity between complete order (periodicity, with in the most extreme case: constancy) and complete disorder (randomness). As the seeds in the IGIP category are infinite in size, it is possible to consider the extreme cases for order and disorder in the seed-configuration (practically this requires that the seed be sufficiently large). We will present here a few examples that show how genotypes of both extreme types of complexity create phenotypes with intermediate complexity (of the quasiperiodic type).

This is about complex three-dimensional objects. Within the scope of this paper, it is only possible to produce but a rather fragmentary image, through a combination of pictures and words, of the structure of a few selected CGI-orbits. Nevertheless, we will try to convey a few general aspects of possible behavior.

5.1. *IGIP-C examples*

Example 5. It concerns the CGI-problem described in Appendix A, item (b). It has the collinear renormalizing neighborhood of Fig. 10(b). The shift-parameters are $a_{\sigma_1,\sigma_2} = b_{\sigma_1,\sigma_2} = 0$ (for all $(\sigma_1, \sigma_2) \in \{0, 1\}^2$). The seed is taken to be unidirectional along the $/$-diagonal, and is given by $\{a(k, -k+l) | k \in \mathbb{Z}, -k+l = \lambda = -2, -1, 0, 1, 2, 3\}$. All seedcells are freecells.

Example 6. This deals with the CGI-problem described in Appendix A, item (e). It has the collinear renormalizing neighborhood of Fig. 10(e). The shift-parameters are $a_{\sigma_1,\sigma_2} = b_{\sigma_1,\sigma_2} = 0$ (for all $(\sigma_1, \sigma_2) \in \{0, 1\}^2$). The seed is unidirectional along the $/$-diagonal, and is given by $\{a(k, -k+l) | k \in \mathbb{Z}, -k+l = \lambda = -2, -1, 0, 1, 2\}$. All seedcells are freecells.

Figure 16 shows the initial configuration solutions for Example 5, with a randomly filled seed (a), a quasiperiodic seed (c), and a constant seed [(e), all seedcells are in state 1]. Figures 16(b), 16(d) and 16(f) show the solutions for Example 6, given

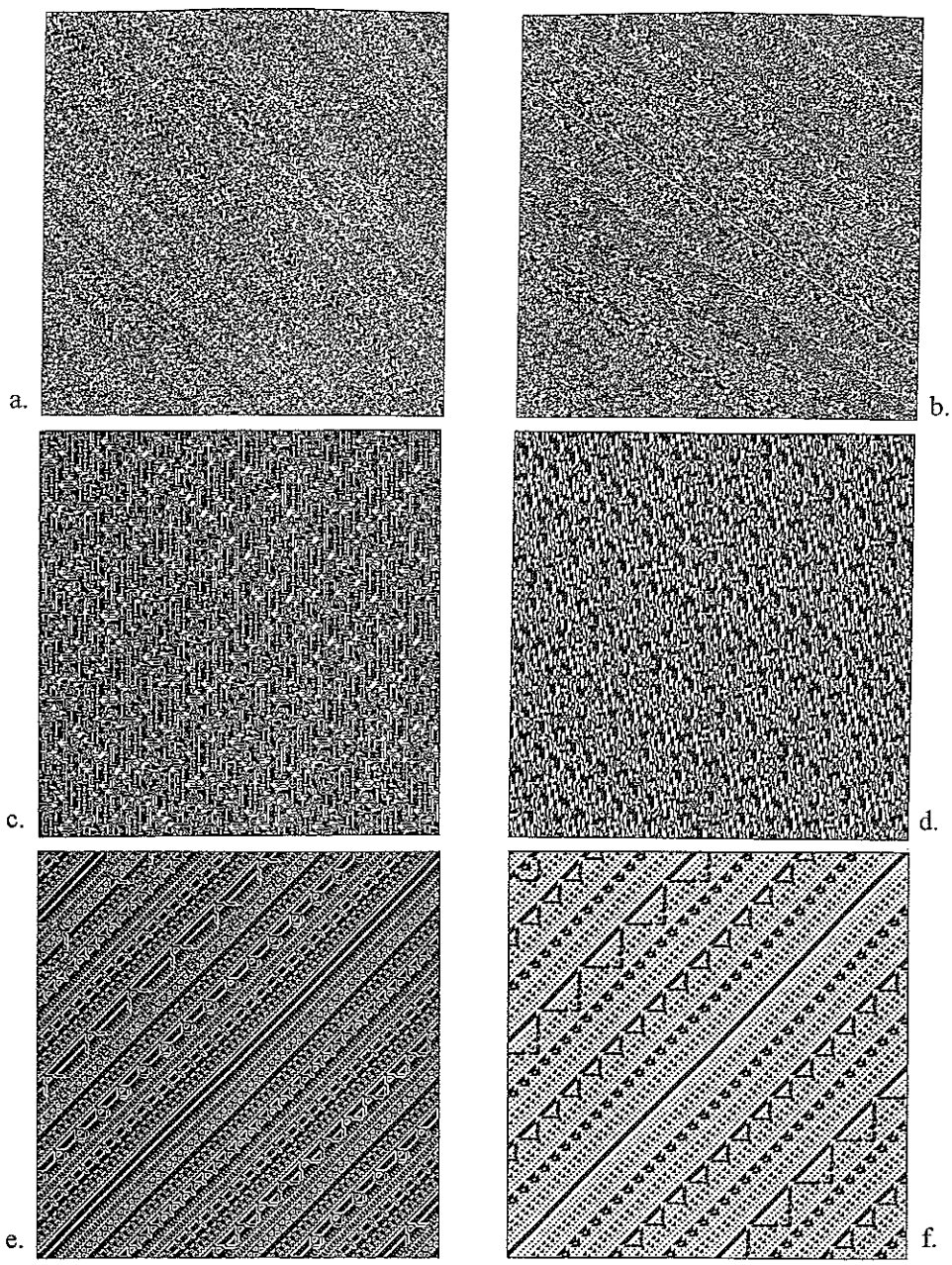


Fig. 16. Initial configuration solutions for IGIP-C CGI-problems. (a), (c), (e): Example 5 problem. (a) Random seed; (c) quasiperiodic seed; (e) seed cells are all in state 1. (b), (d), (f): Example 6 problem. (b) Random seed; (d) quasiperiodic seed; (f) seed cells are all in state 1.

the same type of seed configurations (see the Appendix B for the description of the quasiperiodic seeds). Observe, for the random seed, the appearance of a dominant orientation in the patterns along the direction defined by the collinear renormalizing neighborhood. The sequences along lines in this direction are quasiperiodic (in the sense discussed in [Barbé *et al.*, 1995; Barbé, 1996a]: strings of any size in these sequences are recurrent, but in a seemingly irregular way). This is also true for the solutions with quasiperiodic and constant seed. The structure along lines in the $\diagup$-direction, which is also the seed-direction, inherits the complexity class of the seed (random, periodic, quasiperiodic).

The quasiperiodicity of these patterns along the renormalizing neighborhood direction is made more manifest by considering the scaling properties of some of its *shift-add correlation patterns* (SAP's),

as was established in [Barbé, 1996b] [see also [Barbé *et al.*, 1995] and [Barbé, 1996a], where it was introduced first and was called shift-identity pattern (SIP)]. For convenience, we recall here that a (v, h)-SAP of a two-dimensional pattern is the pattern obtained by pixelwise addition (modulo 2 in the $\mathbb{Z}_2$-case considered here) of the original pattern with a copy of it that has been shifted vertically over v pixels, and horizontally over h pixels. This will be indicated as a (v, h)-SAP.

Such SAP's are shown in the left column of Fig. 17 for the random-seed initial configuration of Fig. 16(a), and in the right column of Fig. 17 for the random-seed initial configuration of Fig. 16(b). A white pixel in these SAP's indicates where the original pattern and its (v, h)-shifted version have the same pixel-value; a black pixel indicates where the pixel values are different. Observe that the $(8, 8)$-, the $(16, 16)$- and the $(32, 32)$-SAP's in the left column of Fig. 17 exhibit scaling: the linear

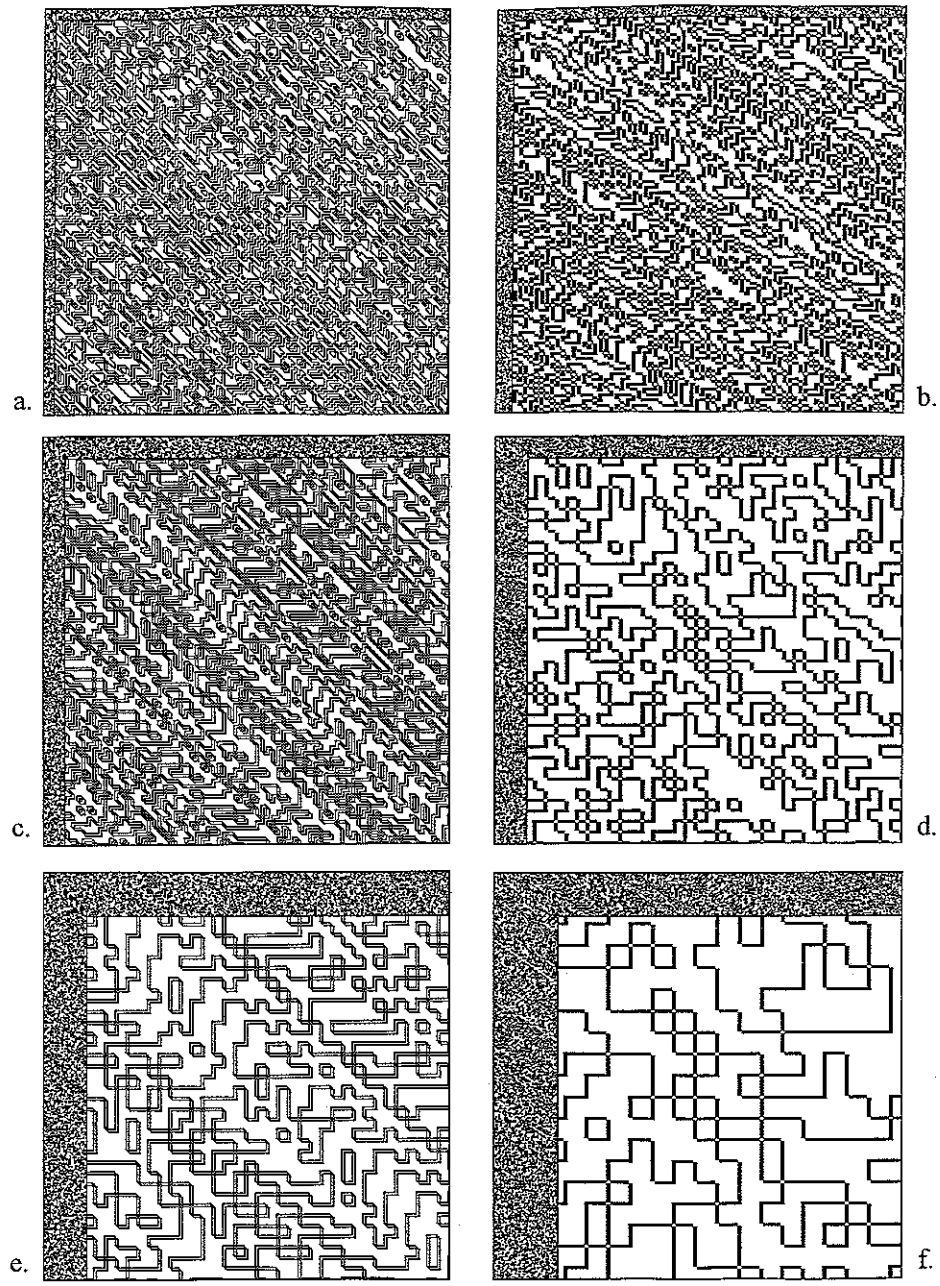


Fig. 17. (a), (c), (e): Shift-add correlation patterns (SAP's) for the initial configuration of Fig. 16(a) [(a) $(v, h) = (8, 8)$; (c) $(v, h) = (16, 16)$; (e) $(v, h) = (32, 32)$]. (b), (d), (f): SAP's for the initial configuration of Fig. 16(b) [(b) $(v, h) = (8, 12)$; (d) $(v, h) = (16, 24)$; (f) $(v, h) = (32, 48)$].

size of the white domains seem to double. Actually, Fig. 17(e) is a kind of $\times 2$-enlarged version of Fig. 17(c). This scaling behavior seems to occur for all $(2^i, 2^i)$-SAP's over increasing i, starting from some initial $i_0 \neq 0$. Moreover, we have also observed that the SAP's for $(v, h) = (\sum_{i_{\min}}^{i_{\max}} 2^i, \sum_{i_{\min}}^{i_{\max}} 2^i)$, [with $i_{\min} \geq i_0$], show a similar pattern as the $(2^{i_{\min}}, 2^{i_{\min}})$-SAP. We call this kind of scaling: *Minimal power scaling* (MP-scaling). For example, the (v, h)-SAP's of Fig. 16(c) with $(v, h) = (24, 24), (40, 40), (72, 72)$, or in general $(\sum_{i=3}^{i_{\max}} 2^i, \sum_{i=3}^{i_{\max}} 2^i)$, have a similar structure as the $(8, 8)$-SAP shown in Fig. 17(a); the $(48, 48)$-, $(80, 80)$-, ..., $(\sum_{i=4}^{i_{\max}} 2^i, \sum_{i=4}^{i_{\max}} 2^i)$-SAP's have a similar structure as the $(16, 16)$-SAP displayed in Fig. 17(c).

So, for increasing i, the $(2^i, 2^i)$-SAP's show white domains that double in linear size: This means that the original pattern and its shifted version have identical subpatterns of increasing size along the dominant $\backslash$-diagonal direction. Notice that the inflation of the white domains occurs in the two dimensions of the pattern, not only in the direction of that diagonal shift. This is even more striking through the observation that pure vertical

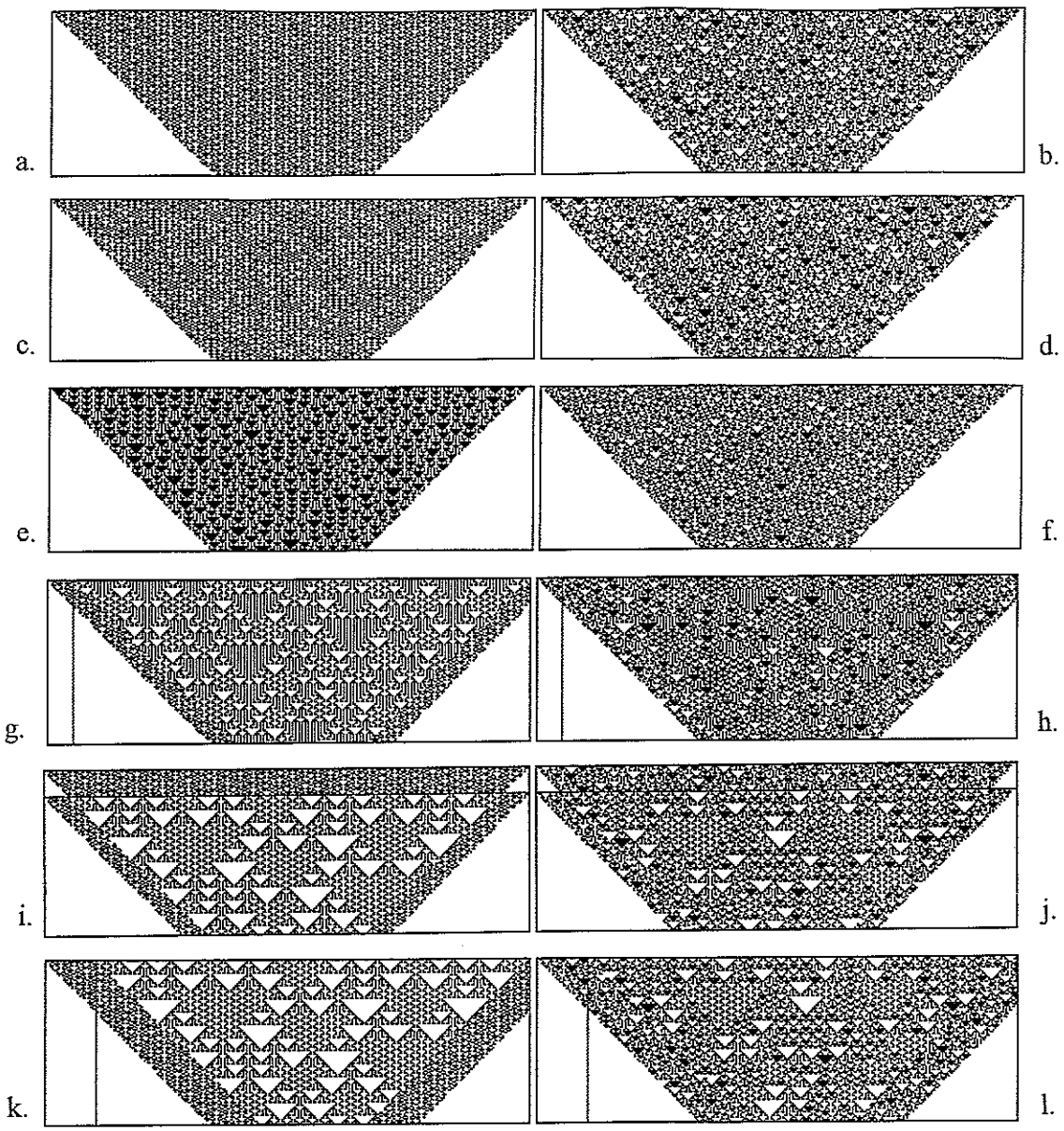


Fig. 18. Q-slices for the orbit with initial configuration from Fig. 16(a) (random seed). (a)–(f): Q_0, Q_1, ... to Q_5, (in that order). (g), (i), (k): SAP's for Q_0 [shown under (a)]: (g) $(v, h) = (0, 16)$; (i) $(v, h) = (16, 0)$; (k) $(v, h) = (0, 32)$. (h), (j), (l): The same SAP's for Q_1 [shown under (b)].

or pure horizontal shifts, or (v, h)-shifts in which (v, h) is not of the type mentioned above, produce SAP's which are similar to the original pattern or which look even more random.

Similar properties and observations exist for the patterns in the second column of Fig. 17. The only difference is the dominant direction [slope $\frac{2}{3}$ in the (k, l)-plane]: MP-scaling occurs here for shifts $(v, h) = (\sum_{i=i_{\min}}^{i_{\max}} 2^i, \frac{3}{2} \sum_{i=i_{\min}}^{i_{\max}} 2^i)$.

It is also interesting at this point to compare the full CGI-solution to the solution of the corresponding equation without rescaling term, and starting from the same seed. That "without rescaling term"-counterpart of Fig. 16(a) is displayed in Fig. 22(a). All sequences along the dominant direction are *periodic*, either with period 1, 2 or 4 (this is easily linked to the particular structure of the renormalizing neighborhood and the way of propagating the solution in this case).

Figures 18(a)–18(f) display some Q-slices of the orbit generated by the random-seed configuration of Fig. 16(a). The patterns Q_0, Q_2, Q_4 (and Q_{2i} in

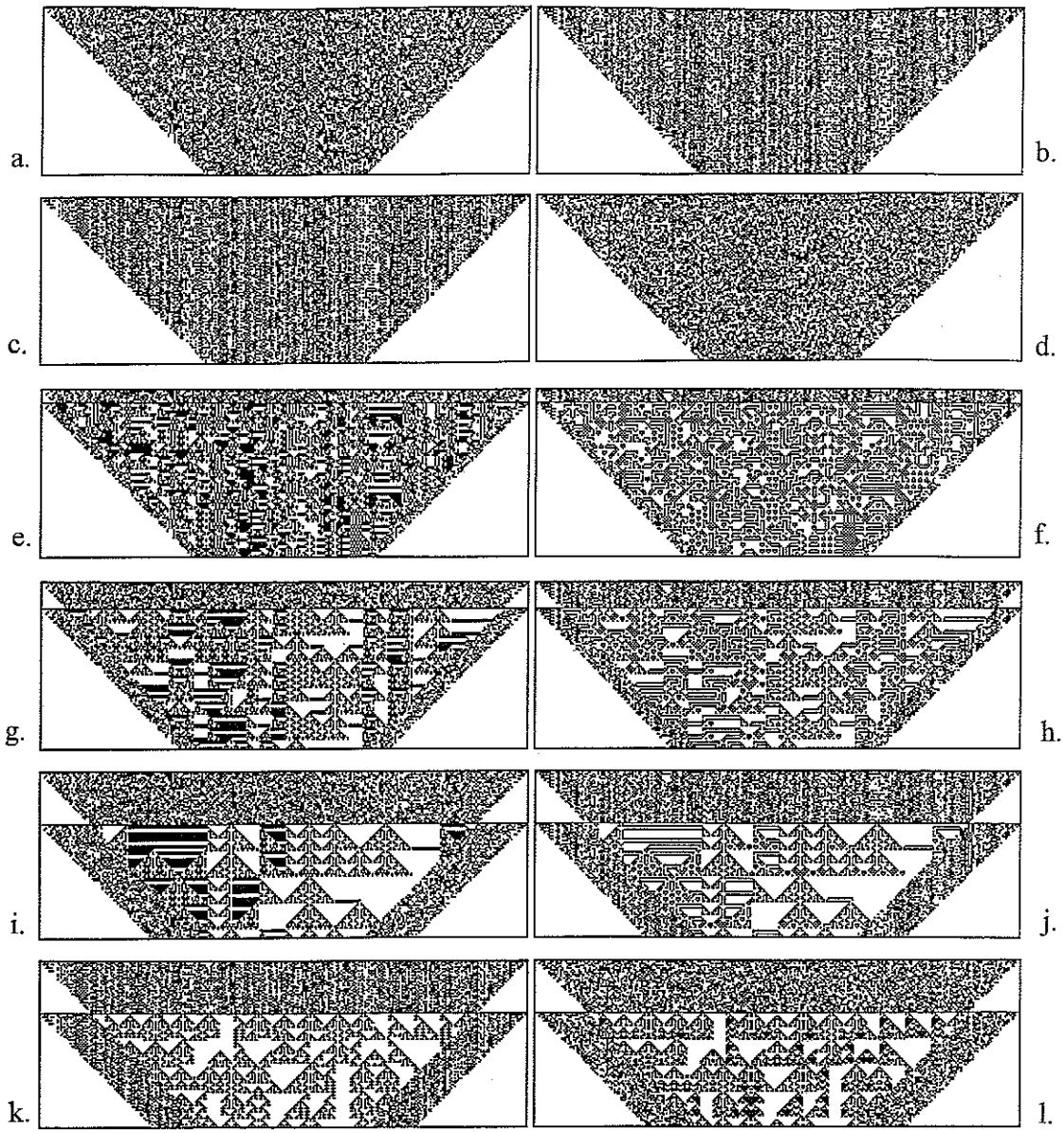


Fig. 19. L- and D-slices of the orbit with initial configuration of Fig. 16(a) (random seed). (a), (b): L_0 and L_1; all L_{2i} are similar to L_0, all L_{2i+1} are similar to L_1. (c), (d): D_0 and D_1; all D_{2i} are similar to D_0, all D_{2i+1} are similar to D_1. (e), (g), (i): SAP's of L_0 [displayed under (a)] over vertical shifts 8, 16, 32 respectively. These are scaling. (f), (h), (j): The same SAP's for L_1 [displayed under (b)]. (k), (l): $(32, 0)$-SAP's for the D_0- and D_1-patterns in (c) and (d).

general) are reminiscent of the quasiperiodic structures that appeared in [Barbé *et al.*, 1995] and in [Barbé, 1996a]; the patterns Q_1, Q_3, Q_5 (Q_{2i+1} in general) are reminiscent of the patterns in the "complex category" presented in the last mentioned reference. The remaining patterns in Fig. 18 are SAP's: They are MP-scaling for $(0, \sum 2^i)$ and $(\sum 2^i, 0)$-shifts. Observe the peculiarity that the $(16, 0)$-SAP's and the $(0, 32)$-SAP's [Figs. 18(i) and 18(k)] seem to be identical!

Figure 19 shows L- and D-slices for Fig. 16(a), and also some of their SAP's. These SAP's are MP-scaling for $(v, h) = (\sum 2^i, 0)$-shifts.

Figure 20 shows a few SAP's for the quasiperiodic-seed and constant-seed initial configurations of Figs. 16(c) and 16(e). In addition to the

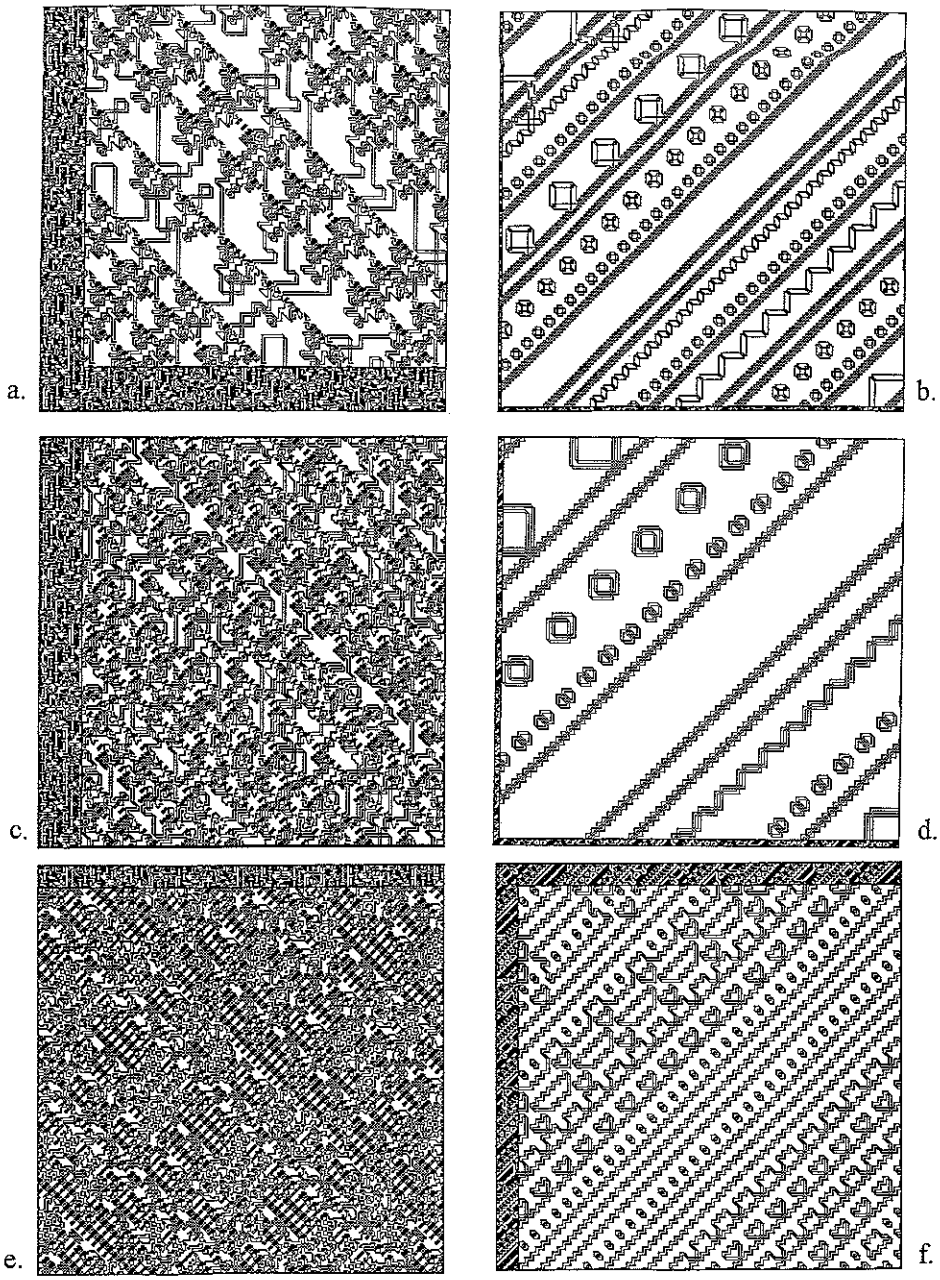


Fig. 20. (a), (c), (e): SAP's for the initial configuration of Fig. 16(c) (with quasiperiodic seed); [(a) $(v, h) = (-32, 32)$; (c) $(v, h) = (0, 32)$; (e) $(v, h) = (16, 0)$]. (b), (d), (f): SAP's for the initial configuration of Fig. 16(e) (with constant seed); [(b) $(v, h) = (-2, -2)$; (d) $(v, h) = (-4, 4)$; (f) $(v, h) = (16, 16)$].

MP-scaling for $(\sum 2^i, \sum 2^i)$-SAP's which also existed for random-seed solutions, there is now also MP-scaling for $(0, \sum 2^i)$-, $(\sum 2^i, 0)$-, $(-\sum 2^i, \sum 2^i)$-SAP's. Observe that, for the case with constant seed, the white band along to the $/$-diagonal seed direction doubles in (linear) size with each step in the $(-2^i, 2^i)$- SAP sequence. This indicates that the original pattern is periodic along lines parallel to this seed, with period doublings occurring when one moves further away from the seed in a direction orthogonal to it, while the larger periods remain constant over wider bands (twice as wide bands for twice as large periods).

Figure 21 shows a few Q, L and D-slices and related SAP's for the orbit obtained from the constant-seed solution in Fig. 16(e). SAP's for the L-slices now also exhibit MP-scaling for $(0, \sum 2^i)$-shifts (which was not so for the L-slices in the random-seed case). Observe the nice frieze-like patterns in the D-slices. They are periodic in the horizontal direction, with period-doublings occurring over ever-doubling distances when moving

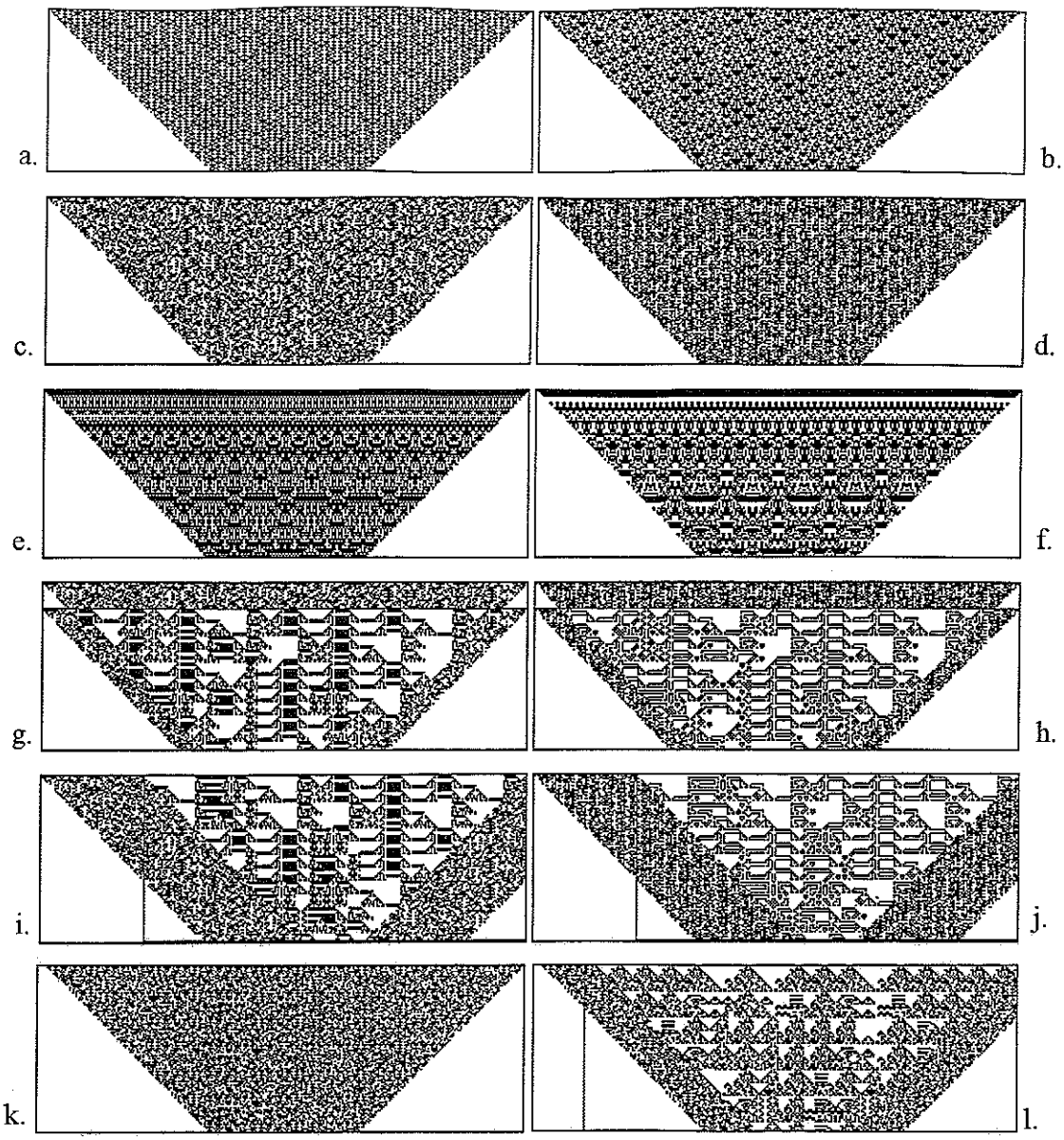


Fig. 21. (a)–(j): Orbit slices and SAP's related to the orbit generated by Fig. 16(e) (with constant seed) (a), (b): Q_2- and Q_3-slices (quasiperiodic patterns); (c), (d): L_0- and L_1-slices (quasiperiodic); (d), (f): D_0 and D_1-slices (horizontally periodic, period doublings). (g), (h): $(16, 0)$-SAP's of the patterns in (c) and (d); (i), (j): $(0, 64)$-SAP's of the patterns in (c) and (d). (k): D_5-slice for the orbit generated from Fig. 16(c) (quasiperiodic seed); (l): The $(0, 32)$-SAP of (k).

downwards. Figure 21(k) shows a D-slice of the orbit obtained from the quasiperiodic-seed solution in Fig. 16(c). It has a complexity that lies somewhere in between the complexities of the D-slice patterns for the random-seed and constant-seed solutions. It has also MP-scaling for $(0, \sum 2^i)$- SAP's [see Fig. 21(l)].

Figure 22 shows a few more patterns associated with the solution of the corresponding CGI-equations without rescaling term. They should be compared with their full CGI-counterparts.

The fact that there are such nice quasiperiodic or complex Q-slices for the orbits related to Example 5 has certainly to do with the fact that the CA-rule for this case is also collinear and acts along lines oriented in the Q-direction (which is here also the dominant renormalizing neighborhood direction). This implies that the Q-slices are obtained as the orbit of a one-dimensional CA. The corresponding one-dimensional initial configurations are the separate sequences along the $\backslash$-diagonal direction in Fig. 16(a), the one-dimensional CA-rule is $R(X) = 1 + X + X^{-1}$ (the original two-dimensional rule being $R(X, Y) = 1 + XY + X^{-1}Y^{-1}$).

The statement in the previous paragraph is also corroborated by the fact that a modification of the CGI-problem of Example 5 such that the renormalizing neighborhood remains the same, while the CA-rule becomes *non-collinear*, changes the outlook of the Q-patterns to a less-structured one. This is what occurs in the next example.

Example 7. The problem parameters are those of Appendix A, item (c). It has the same collinear neighborhood structure as Example 5. The CA-rule is non-collinear: $R(X, Y) = X + XY + X^{-1}Y^{-1}$. The shift-parameters are also the same as those for Example 5: $a_{\sigma_1,\sigma_2} = b_{\sigma_1,\sigma_2} = 0$. The seed remains the same as in Example 5.

The initial configuration solutions for the CGI-problem in Example 7 are very similar to those for Example 6 under the same kind of seeds. Also their SAP's are similar, with the same scaling property. However, the vertical orbit slices are less-well structured, except on the hidden level of their SAP's, which do exhibit MP-scaling. This is shown

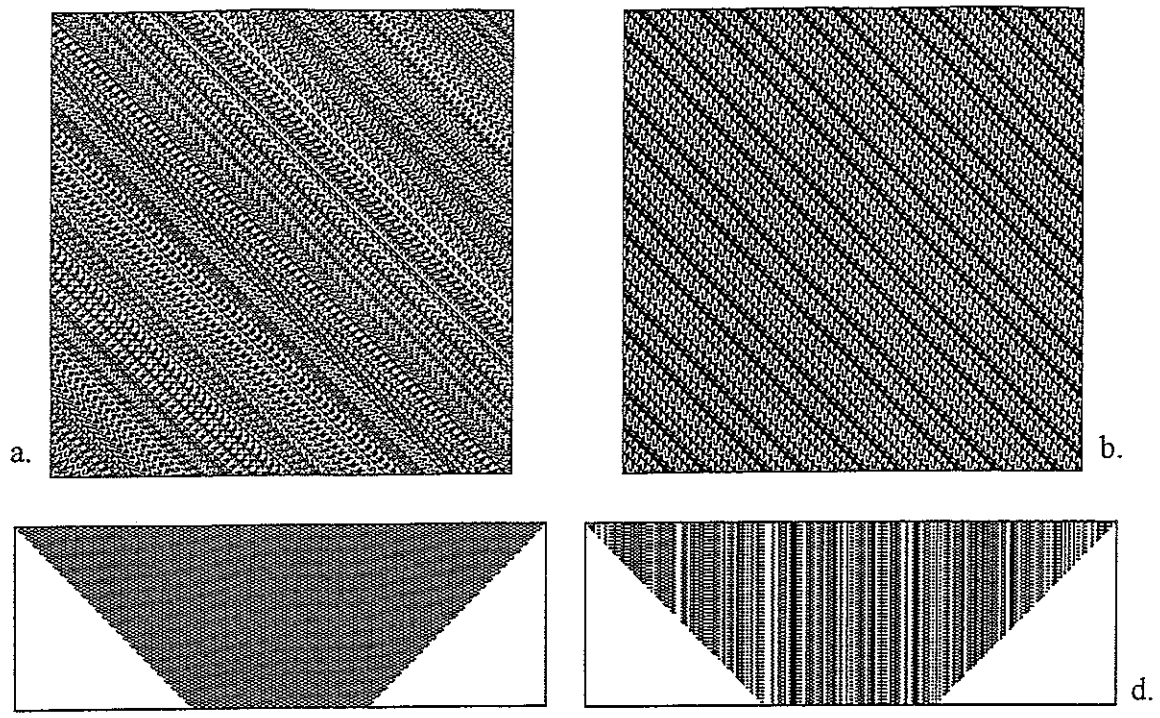


Fig. 22. (a) Solution of the CGI-equation without rescaling term corresponding to Fig. 16(a) (same random seed). (b) Solution of the CGI-equation without rescaling term corresponding to Fig. 16(c) (same quasiperiodic seed). (c) Q_0-slice of the orbit generated by (a) [compare to Fig. 18(a)]. (d) D_0-slice of the orbit generated by (a) [compare to Fig. 19(c)]. All Q_i-slices are similarly homogeneous as Q_0 shown in (c): Or they overall black or white, or have alternating vertical black and white stripes.

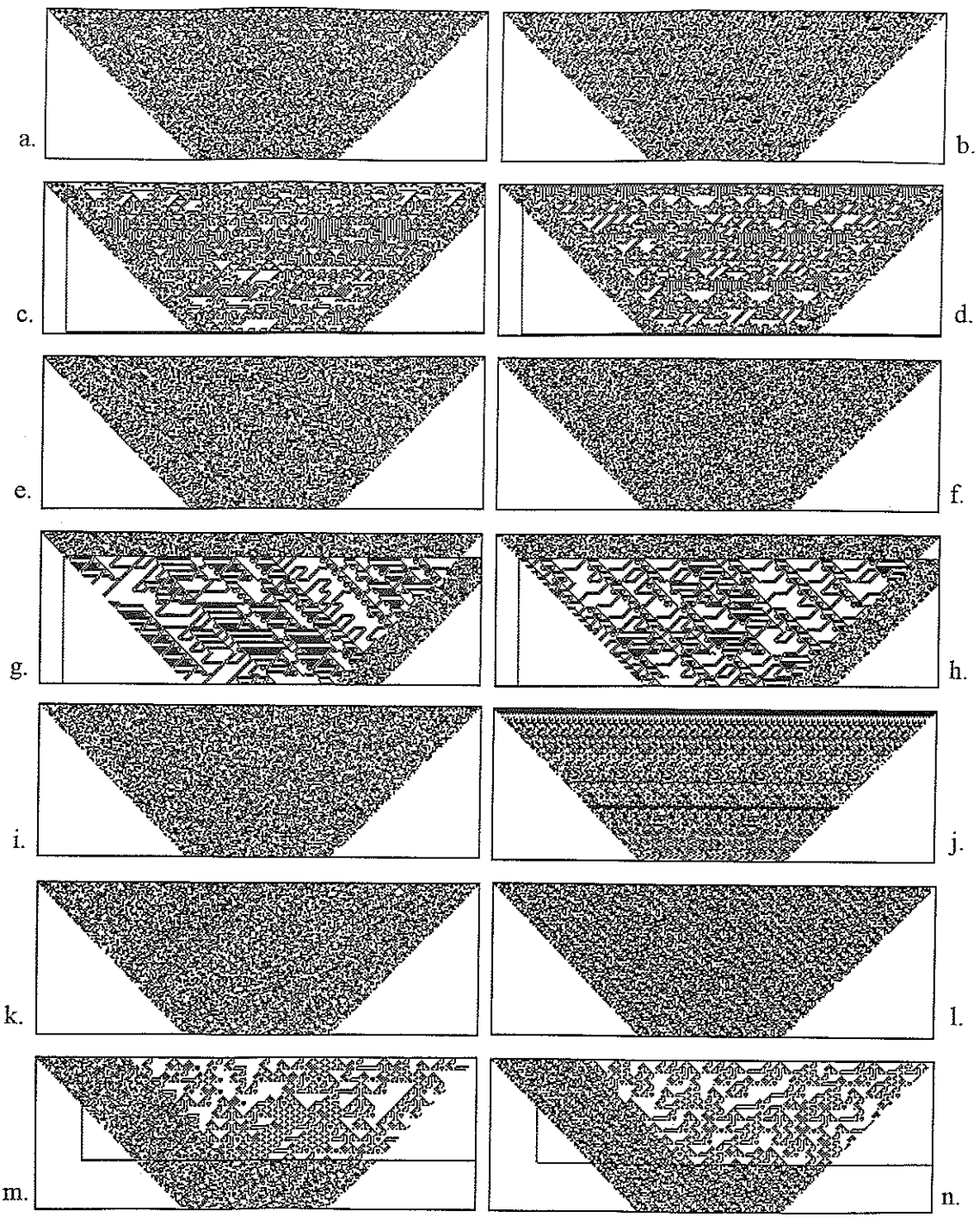


Fig. 23. Orbit-slices for the IGIP-C problem of Example 7, with non-collinear CA-rule. The patterns in the left column correspond to a random seed, those in the right column to a constant seed. (a), (b): Q_0-slices; (c), (d): $(0, 16)$-SAP's of (a) and (b); (e), (f): L_0-slices; (g), (h): $(16, 16)$-SAP's of (e) and (f); (i), (j): D_0-slices; (k), (l): K_0-slices; (m), (n): $(-32, 32)$ SAP's of (k) and (l).

in Fig. 23 for a few orbit slices obtained from the random-seed and constant-seed configuration. These patterns should be compared to the corresponding ones in Figs. 19 and 21. Notice how close the Q_0-and L_0-patterns and their SAP's have become for both random and constant seed. The D_0-slices differ most: they inherit the features of the seed. Also observe the change in dominant orientation between the K-slices in the two cases [Figs. 23(k) and 23(l)].

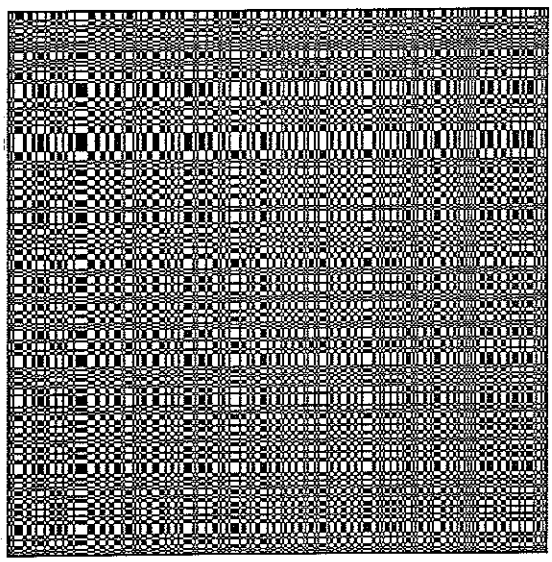


Fig. 24. The solution of the CGI-equation without rescaling term corresponding to Fig. 15(b) (IGIP-nC example with random seed).

5.2. *An IGIP-nC example*

We will consider here the initial state configuration solution and corresponding orbit of the CGI-problem that was solved in Example 4, starting from a bidirectional seed with random freecells and constant freecells. The initial configuration solutions were shown in in Figs. 15(b) and 15(c) respectively.

Let us first discuss the random-seed solution in Fig. 15(b). This pattern looks very unstructured: The dominant orientation that was present in the IGIP-C random-seed example has disappeared now. It is also interesting to compare it with the solution of the corresponding CGI-equation without rescaling term, which is shown in Fig. 24. The rescaling term seems to have destroyed most of the structure that is present in this "without rescaling term"-solution.

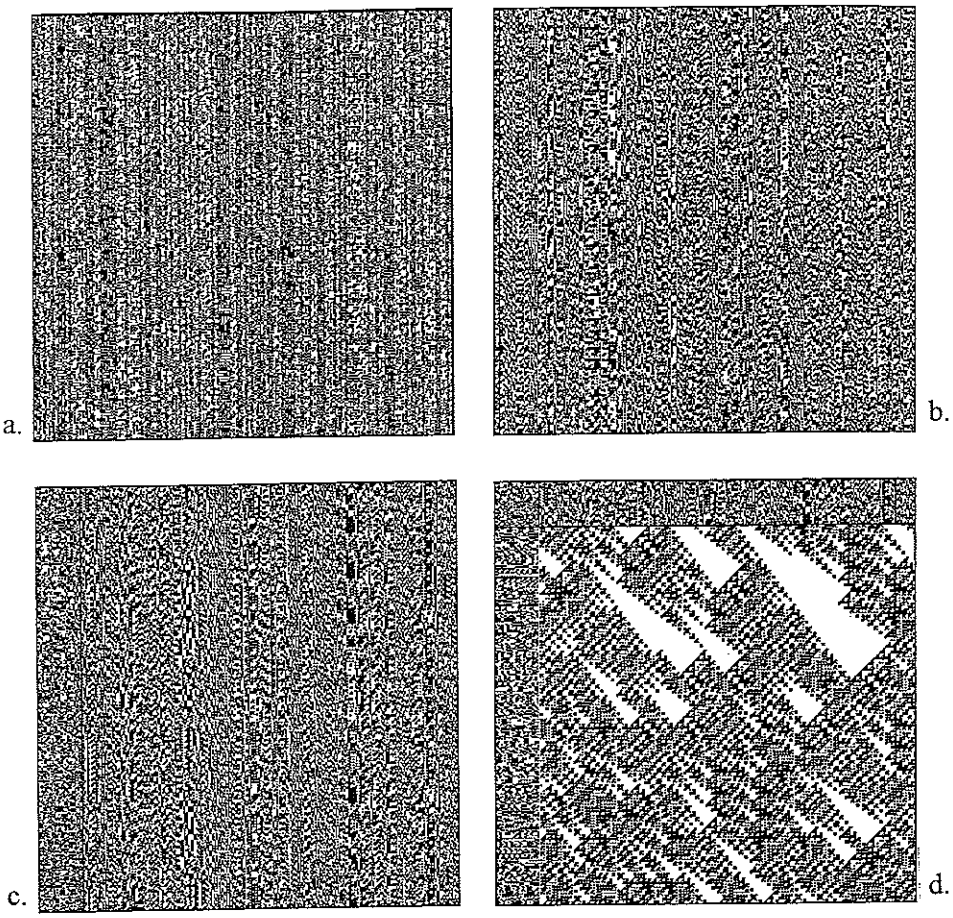


Fig. 25. SAP's of the pattern in Fig. 15(b) (IGIP-nC example with random seed). (a) $(v, h) = (0, 1)$; (b) $(v, h) = (0, 4)$; (c) $(v, h) = (0, 32)$. Similar patterns as the ones in (b) and (c) occur for all $(0, \sum 2^i)$-SAP's (but with proper MP-scaling). (d) a *second order* SAP: The $(32, 0)$-SAP of pattern (c). SAP's for $(v, h) = (1, 0)$, $(4, 0)$, $(32, 0), \ldots (\sum 2^i, 0)$ are similar in nature to those above, but horizontal and vertical features are interchanged. SAP's for other shifts are similar to the original pattern in Fig. 15(b).

Figure 25 shows several SAP's of Fig. 15(b), and these uncover some interesting hidden structure. The $(0, 2^i)$-SAP's display some quasiperiodic behavior in the vertical direction (this has clearly become a dominant orientation in these SAP's). But these patterns are still very intricate. A real *surprise* comes with the *second order* SAP (a SAP of a SAP) which is shown in Fig. 25(d): A pattern emerges that is still very complex, but that consists of well-structured and aesthetically appealing subpatterns on the smaller scales. This pattern is really present in the apparent disorder of Fig. 15(b)! This second order SAP is also MP-scaling for $(v, h) = (\sum 2^i, 0)$.

Figure 26 shows a few slices of the CG-invariant CA-orbit generated from Fig. 15(b) under the rule $R(X, Y) = X + Y + XY$. It was observed that all horizontal T_i-slices are similar to Fig. 15(b). The Q_i, L_i, D_i, K_i-slices all have quasiperiodic structures in the vertical direction. This is corroborated

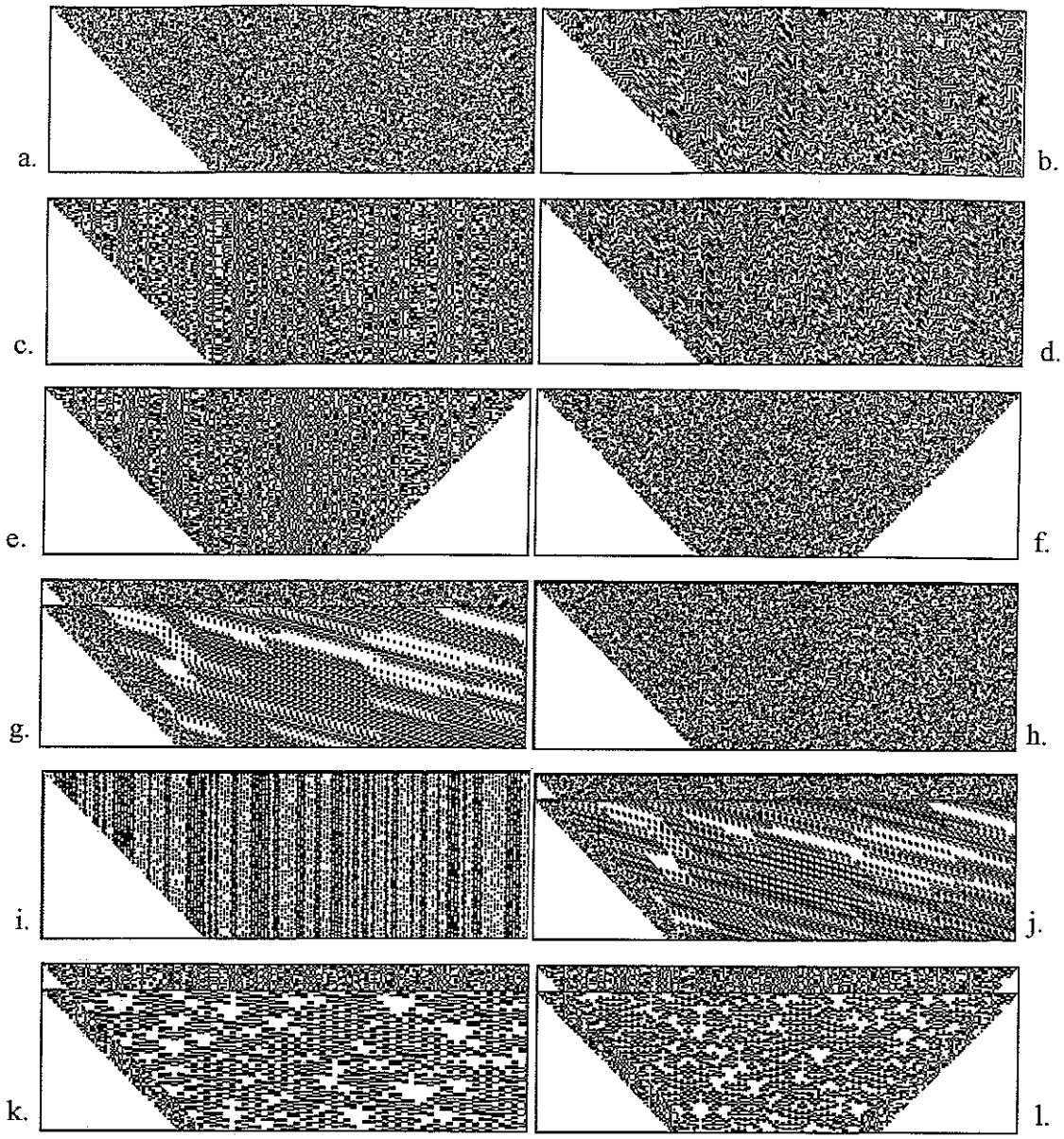


Fig. 26. Slices of the CGI-orbit generated from Fig. 15(b) (IGIP-nC case with random-freecell seed). (a), (b): Q_0- and Q_1-slices. All Q_{2i}-slices are similar to Q_0, all Q_{2i+1} slices are similar to Q_1. (c), (d): L_0- and L_1-slices. All L_{2i}-slices are similar to L_0, all L_{2i+1} slices are similar to L_1. (e), (f): D_0- and D_1-slices. All D_{2i}-slices are similar to D_0, all D_{2i+1} slices are similar to D_1. The K-slices are similar to the L-slices. (g) $(16, 0)$-SAP of (a) (similar for all Q_i); (h) $(1, 1)$-SAP of (a); (i) $(1, 1)$-SAP of (b); (j) Second order SAP, the $(16, 0)$-SAP of (h); (k) $(16, 0)$-SAP of (c); (l) $(16, 0)$-SAP of (e).

by their vertical SAP's, which are MP-scaling for $v = \sum 2^i$.

Let us now contemplate some patterns related to the constant-freecell solution in Fig. 15(c). The solution of the corresponding CGI-equation without rescaling term is an overall black picture: $(a(k, l))_{(k,l)\in\mathbb{Z}^2} \equiv 1$. Figure 27 displays a few SAP's of Fig. 15(c). These show that this solution is quasiperiodic in horizontal direction [the SAP's are MP-scaling for horizontal shifts, as shown in Figs. 27(a) and 27(b)]. However, this solution is periodic in vertical direction, at least for $k < 0$. This can be seen from the MP-scaling vertical white band in the center of the vertical SAP's shown in Figs. 27(c) and 27(d). This band seems to inflate only to the left, while there is only nonperiodic scaling to the right. So, for $k > 2$, the pattern in Fig. 15(c) seems to be quasiperiodic in vertical

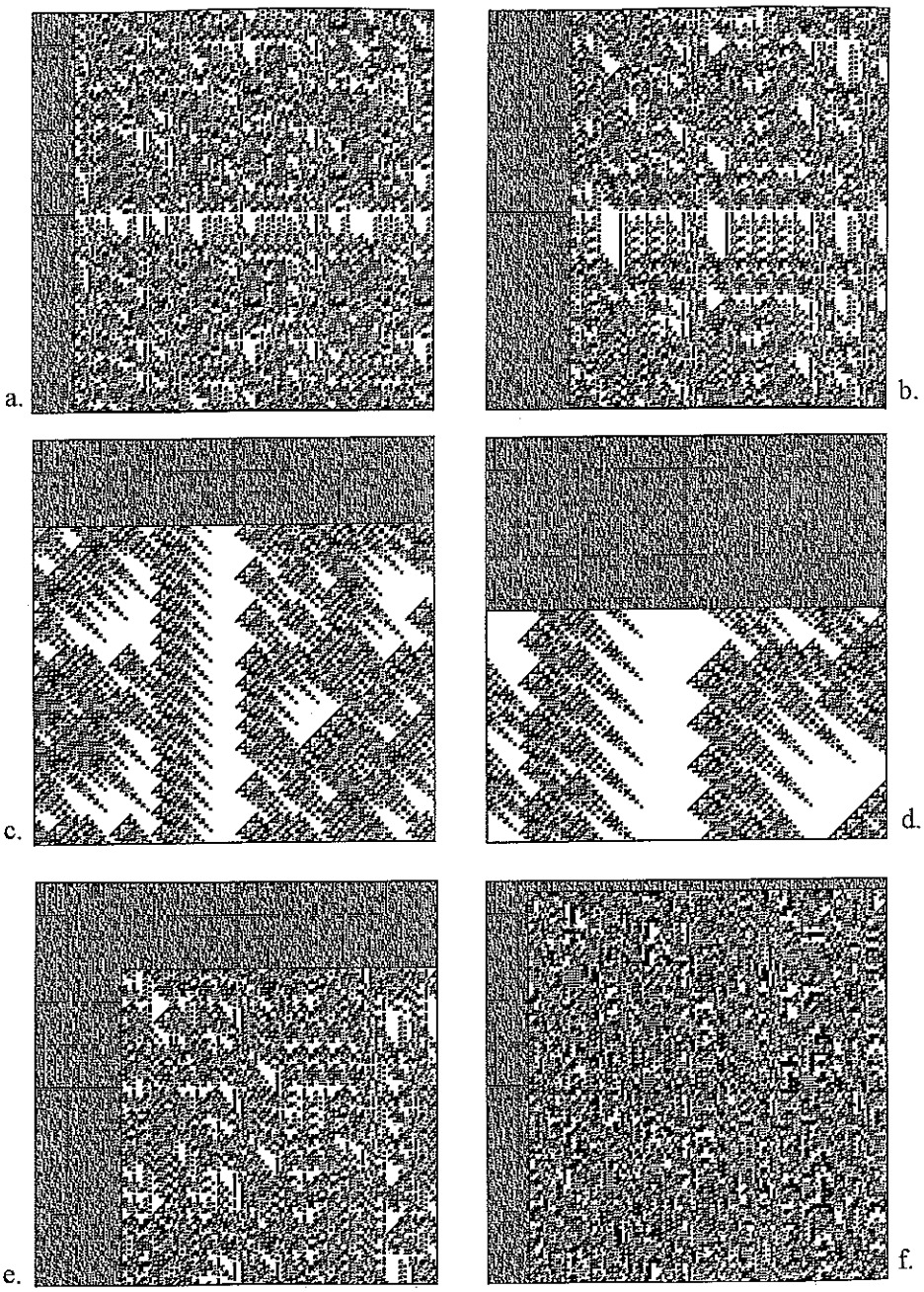


Fig. 27. SAP's of Fig.15(c) (constant-freecell solution of the IGIP-nC Example 4). (a) $(0, 32)$-SAP; (b) $(0, 64)$-SAP, it looks like a $\times 2$-enlargement of (a); (c) $(64, 0)$-SAP; (d) $(128, 0)$-SAP; (e), (f): $(64, 64)$- and $(8, 32)$-SAP's. All $(0, \text{even})$-, $(\text{even}, 0)$-, $(\text{even}, \text{even})$-SAP's are MP-scaling. Compare (c) and (d) to the second order SAP of the random-freecell solution in Fig. 25(d).

direction. This asymmetry is perhaps not so surprising if we recall that the vertical part of the seed too is asymmetric: Only $\{a(-1, l)|l \in \mathbb{Z}\}$ has constant state 1, the remaining vertical part of the seed $(\{a(0, l), a(1, l), a(2, l)| \in \mathbb{Z}\})$ consists of depcells (see Example 4). The way in which these depcells are obtained there reminds of the one-dimensional CGI-case which could indeed produce quasiperiodic sequences [Barbé *et al.*, 1995]. In that way, the pattern in Fig. 15(c) seems to inherit the asymmetric properties of the seed (probably, the limited size of the 2×2-square renormalization neighborhood is also playing a role in this).

It was observed that all ($v = even$, $h = even$)-SAP's of Fig. 15(c) have a structure similar to the ones shown in Figs. 27(e) and 27(f) (but properly MP-scaled). This makes a difference with the random-freecell initial configuration of Fig. 15(b),

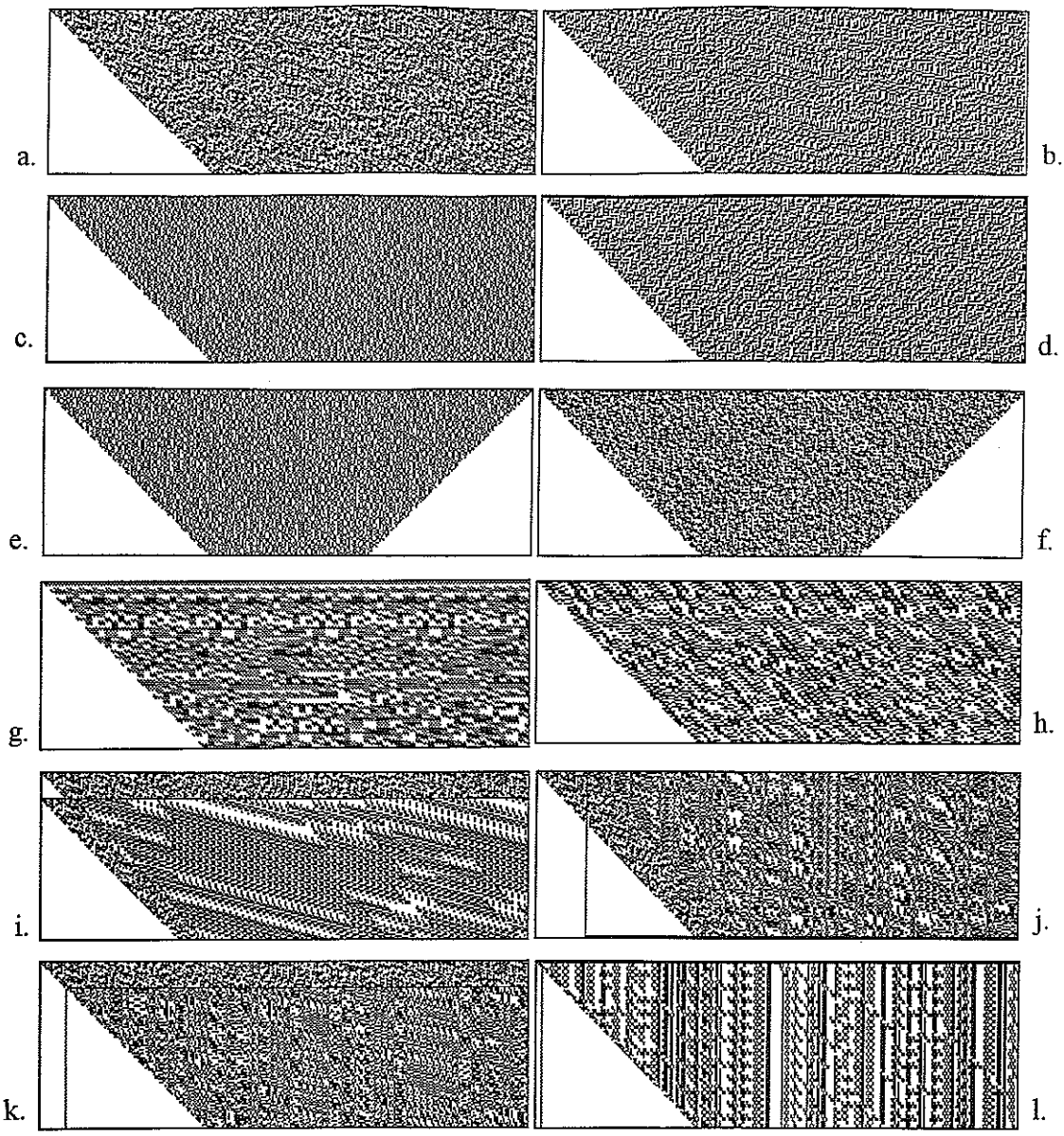


Fig. 28. Slices of the CGI-orbit generated from Fig. 15(c) (IGIP-nC case with constant-freecell seed). (a), (b): Q_0- and Q_1-slices. All Q_{2i}-slices are similar to Q_0, all Q_{2i+1} slices are similar to Q_1. (c), (d): L_0- and L_1-slices. All L_{2i}-slices are similar to L_0, all L_{2i+1} slices are similar to L_1. (e), (f): D_0- and D_1-slices. All D_{2i}-slices are similar to D_0, all D_{2i+1} slices are similar to D_1. (g), (h): K_0- and K_3-slices. Although all K_i-slices are similar in nature, the variability of their visual appearance is largest among them. (i), (j), (k): $(16, 0)$-, $(0, 32)$-, $(16, 16)$-SAP's of Q_0 [i.e. pattern (a)]. (l) $(1, 4)$-SAP of Q_0. All $(k, 4k)$-SAP's are periodic in vertical direction, with increasing complexity as k increases.

for which the SAP's with a shift that was neither stricly vertical nor strictly horizontal, were as unstructured as the original pattern itself.

Figure 28 shows slices of the CGI-orbit generated from Fig. 15(c) through the rule $R(X) = X + Y + XY$. First, it should be mentioned that all horizontal T_i-slices of this orbit have a similar structure as Fig. 15(c) itself (therefore these are not shown). The vertical Q, L, D, K-slices seem quasiperiodic (MP-scaling) in vertical direction. However, there is now also a *more regular structure in other directions*, as can be observed from the MP-scaling (and other) SAP's over non-vertical shifts (apart from a few particular ones for the Q-slices, these SAP's did not show any special structure which was different from the original one in the random-freecell case). Figures 28(i)–28(l) show such SAP's for the Q_0-slice. The vertical (v = even, $h = 0$)-SAP's are similar in nature as those for the random-freecell orbit [compare Fig. 28(i) with Fig. 26(g)]. All $(k, 4k)$-SAP's ($k \in \mathbb{N}$) of the Q_i-slices, like the one in Fig. 28(l), are of a very particular nature: They are periodic in vertical direction [at least in the left half, in the right half they are periodic but possibly

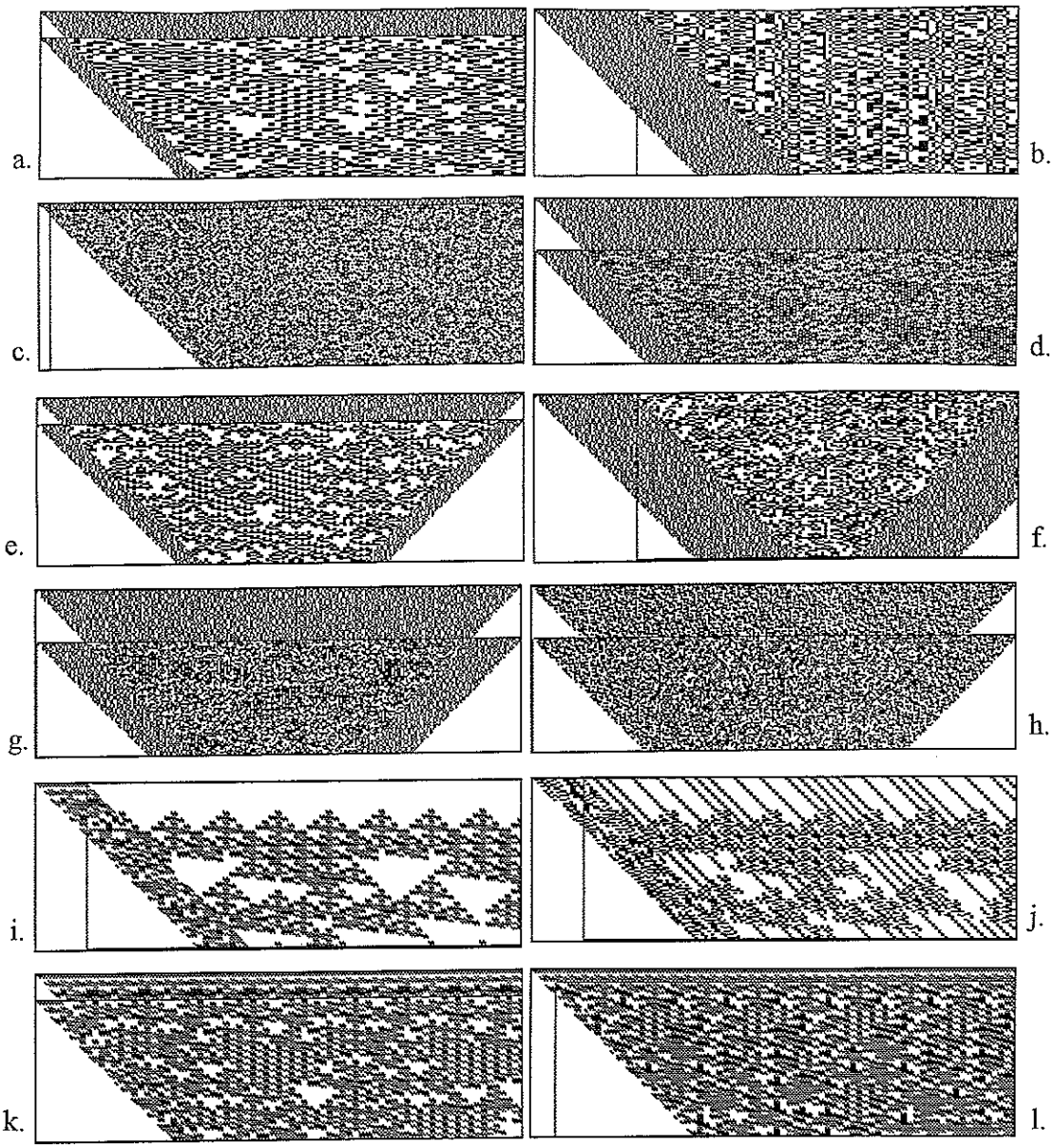


Fig. 29. (a)–(d): $(16, 0)$-, $(0, 64)$-, $(4, 7)$-, $(32, 1)$-SAP's of the L_0-slice in Fig. 28(c). (e)–(g): $(16, 0)$-, $(0, 64)$-, $(32, 1)$-SAP's of the D_0-slice in Fig. 28(e). (h) $(32, 1)$-SAP of the D_1-slice in Fig. 28(f). (i), (j): $(0, 32)$-SAP's of K_0 [Fig. 28(g)] and K_3 [Fig. 28(h)] respectively. (k), (l): $(16, 0)$- and $(8, 16)$-SAP of K_0. Compare (a), (e) with Figs. 26(k) and 26(l).

with some defects (recall the asymmetry in the initial T_0-configuration)]. The displayed $(1, 4)$-SAP is a very simple one (the periodic sequences get more complicated as k grows in the particular $(k, 4k)$-SAP's).

Figure 29 continues with a selection of SAP's of the L, D, K-slices that were shown in Fig. 28. Again, there is MP-scaling in directions for which this was not the case with a random-freecell seed.

Considering the direct outlook of the slices, and taking into account the SAP-properties just mentioned, the orbit of this constant-freecell IGIP-nC example seems to be a *genuine three-dimensional quasiperiodic object*, and it is one of the examples that we think to be close to the three-dimensional equivalent of the two-dimensional quasiperiodic orbits found in [Barbé *et al.*, 1995].

Thus far our attempt to describe some of the complex structures that appear in the solutions of just a few IGIP-C and IGIP-nC type CGI-problems. The most eyecatching difference between the two types was the existence of a dominant orientation coinciding with the orientation of the collinear renormalizing neighborhood for the IGIP-C cases. For both types, random-freecell solutions and constant freecell-solutions produce complex-structured patterns (this complex structure being either directly observable, or through their SAP's). A bit of the randomness, quasiperiodicity, periodicity (constancy) that is present in the seed seems to be inherited in certain aspects of the orbit structure.

5.3. *IGIP-C versus IGIP-nC, once more*

We conclude this section with two examples that once more illustrate the basic distinction between the IGIP-C and IGIP-nC cases. Given, for each case, two seeds that are identical in some kernel-containing finite subset, but different outside that subset, the corresponding initial configuration solutions are identical over an *infinite* band (with finite

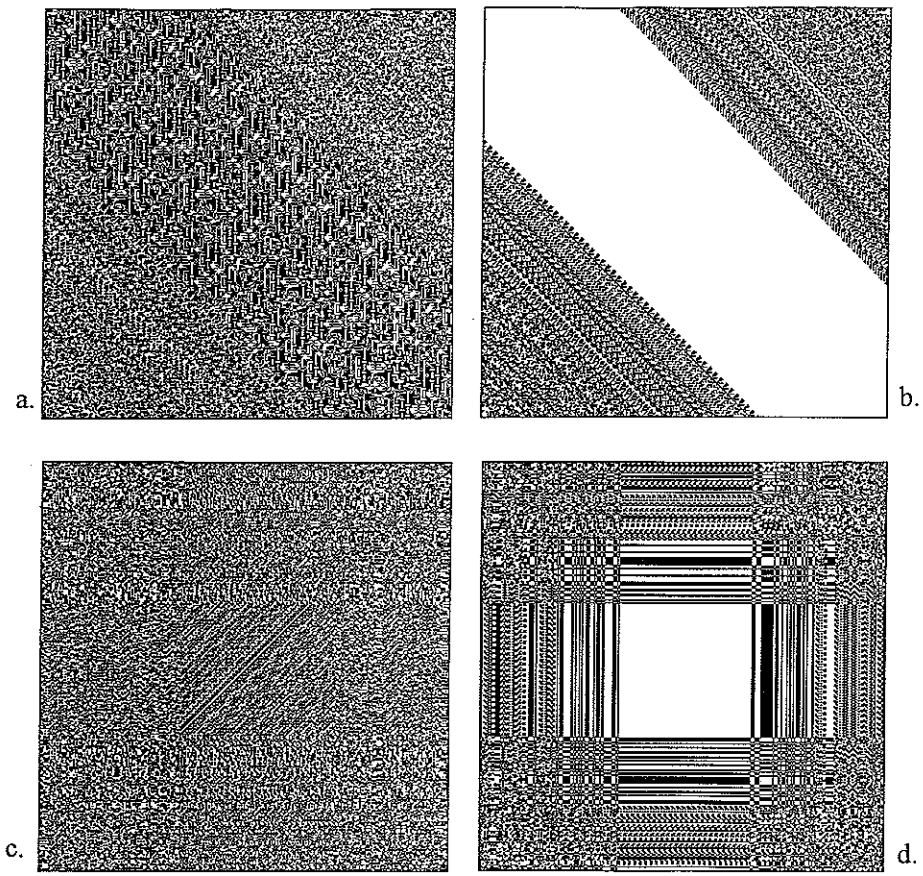


Fig. 30. (a) IGIP-C solution whose seed is partially identical to the one of Fig. 16(c). (b) Pixelwise addition (modulo 2) of (a) and Fig. 16(c). The white area indicates where both patterns are identical. (c) IGIP-nC solution whose seed is partially identical to the one of Fig. 15(a). (d) Pixelwise addition (modulo 2) of (c) and Fig. 15(a).

width) for the IGIP-C case, and identical over a *finite* domain for the IGIP-nC case. This is shown in Fig. 30. Figure 30(a) displays a solution of Example 5 whose seed $\{a_\lambda(k)|k \in \mathbb{Z},\ \lambda = -2,\ -1,\ 0,\ 1,\ 2,\ 3\}$ is identical with that of Fig. 16(c) for $-50 \leq k \leq 50$, but random outside this central part. Figure 30(b) shows the pixelwise addition (mod 2) of the solutions in Fig. 30(a) and Fig. 16(c). The white diagonal band shows where both solutions are identical. This band extends without limits to the upper-left and right-downward diagonal direction. Something similar is done for the IGIP-nC solution in Fig. 15(a). Figure 30(c) shows a solution whose seed is identical with that of Fig. 15(a) inside the confinement $-50 \leq k \leq 50$, and random outside that central part. The addition (mod 2) of the two patterns shows that both solutions are identical in a finite region (the restricted seed solution domain of the identical seed part).

This throws some other light on the nature of the IGIP solutions. The one-dimensional initial configuration solutions of the CGI-problem discussed in [Barbé *et al.*, 1995] were shown to be 2-automatic sequences, obtainable from a finite seed. p-Automaticity of a sequence means that the value of an element in the sequence can be obtained from a finite automaton whose input is the p-adic representation of the address of that element. This concept can be extended to higher-dimensional sequences [Salon, 1986], possibly using more exotic numeration systems [Allouche *et al.*, 1995]. There are indications that p-automaticity holds for all one-dimensional CGI-problems over $\mathbb{Z}_p$ [Skordev, private communication]. However, for two-dimensional IGIP-type problems, *most* of the coarse-graining invariant initial configuration solutions *cannot* be automatic. The reason is that automatic sequences form a denumerable set, while the set of IGIP seed-configurations that generate different CGI-solutions is non-denumerable because of the infinite number of freecells in the seed. Indeed, a finite (kernel-containing) seed-restriction only generates an incomplete part of the solution: Adding one layer to an already existing part of the solution requires fixing of at least one additional freecell in the seed. Automaticity will only be possible for carefully selected seed-configurations [perhaps those with automatic subsequences like the ones displayed in Figs. 16(c)–16(f)]. We conjecture that there might be some weaker form of two- or three-dimensional automaticity in the three-dimensional IGIP-orbits, such as automaticity for certain lower-dimensional subsequences (along the dominant orientation for the IGIP-C examples). The MP-scaling properties of the SAP's could be an indication for this. But this needs a more profound investigation.

6. Conclusions

The three-dimensional coarse-graining invariant orbits of two-dimensional CA's that were studied in this paper show similarities as well as dissimilarities with the two-dimensional CGI orbits of one-dimensional CA's (studied in [Barbé *et al.*, 1995] and [Barbé, 1996a]).

The similarities concern the complex patterns, i.e. patterns of an intricate nature in between simple order and complete disorder, in these orbits. Also the principles for solving the underlying CGI-equations, which have the form of renormalizing difference equations, are the same in both cases: The solution can be propagated in a recursive way, starting from some seed that has to be determined first. The practice of doing this is however more cumbersome in the higher-dimensional case.

The main difference between the one- and two-dimensional CA-cases lies in the nature of the seed (genotype) that produces the whole solution (phenotype). Only a minority of the two-dimensional CGI-problems (the FGIP-problems) leads to a seed with finite supporting core, while this is always the case for the one-dimensional counterpart. Most two-dimensional CGI-problems lead to a seed with infinite supporting core, and can be divided into two subclasses: The IGIP-C and IGIP-nC classes, dependent on whether a finite restriction of the seed can generate an infinite (though restricted) subset or only a finite subset of the solution. The infinite number of freecells in the seed for IGIP-problems also prevents the whole two-dimensional initial configuration solution to be automatic in almost all cases (whereas automaticity seems to be the rule in the one-dimensional case).

It leaves us with an unsatisfactory feeling that the image concerning the complexity of some CGI-orbits in Sec. 5 had to be so fragmentary, although it was sufficient to make clear that three-dimensional orbits obtained from seeds whose freecells have complete random states or simple-order states may exhibit "complex" order. Coarse-graining invariance seems to produce complex order, at least for some class of problem parameters (unfortunately, for practical reasons, our exploration space is too small

for the time being in order to give an assessment of the generality of this statement). Some of this order is rather hidden, but may be uncovered by contemplating shift-add correlation patterns, which have particular scaling properties in many cases. Thereby, we have only considered SAP's of two-dimensional orbit slices.

There are two paths that need to be followed further in order to get a better understanding of the complexities involved. First, the exploration space needs to be enlarged substantially. This requires the necessary software for solving CGI-equations in general, and also visualization tools that allow easy travelling in large areas of three-dimensional spaces (at least $1000 \times 1000 \times 1000$ voxels) and construction of three-dimensional SAP's in a split second. The second path is the analytical path, where one should look for more fundamental properties of the emerging complexities, such as automaticity and the dependency between the problem-parameters and different kinds of complex behavior.

Acknowledgments

The author is grateful to G. Skordev for stimulating discussions and comments. Thanks also to B. Voorhees for his comments on a first draft of this paper.

Part of this work has been carried out within the framework of a Concerted Action Project of the Flemish Community, entitled *Model-based Information Systems* GOA-MIPS.

References

Allouche, J.-P., Cateland, E., Peitgen, H.-O., Skordev, G. & Shallit, J. [1995] "Automatic maps on a semiring with digits," *Fractals* **3**(4), 663–677. Also in *Fractal Geometry and Analysis*, eds. Evertsz, C., Peitgen, H.-O. & Voss, R. F. (World Scientific, Singapore, 1996), pp. 259–273.

Allouche, J.-P., von Haeseler, F., Peitgen, H.-O. & Skordev, G. [1996] "Linear cellular automata, finite automata and Pascal's triangle," *Discrete Applied Mathematics* **66**, 1–22.

Barbé, A., von Haeseler, F., Peitgen, H.-O. & Skordev, G. [1995] "Coarse-graining invariant patterns of one-dimensional two-state linear cellular automata," *Int. J. Bifurcation and Chaos* **5**(6), 1611–1631.

Barbé, A. [1996a] "Coarse-graining invariant orbits of one-dimensional $\mathbb{Z}_p$-linear cellular automata," *Int. J. Bifurcation and Chaos* **6**(12A), 2237–2297.

Barbé, A. [1996b] "Shift-add correlation patterns of orbits of one-dimensional linear cellular automata," in *Int. J. Bifurcation and Chaos* **6**(12B), 2507–2530.

Langton, C. (ed.) [1989] *Artificial Life* (Addison-Wesley).

Martin, O., Odlyzko, A. & Wolfram, S. [1984] "Algebraic properties of cellular automata," *Commun. Math. Phys.* **93**, 219–258.

Salon, O. [1986] "Suites automatiques à multi-indices," *Séminaire de Théorie des Nombres de Bordeaux* **1986–1987**, 4.01–4.27.

Appendix A
CGI-Problem Conditions Generating the Renormalization Neighborhood Examples of Figure 10

All cases considered here concern $\mathbb{Z}_2$-cellular automata and CGI-problems with $2 \times 2 \times 2$ tiles.

(a) Cubic tile ($\Xi \equiv 0$), rule $R(X, Y) = 1 + X + X^{-1}$
$w(0, 0, 0) = w(0, 1, 0) = w(1, 0, 0) = w(1, 1, 0) = 1$, all other $w(r, i, j) = 0$.
Renormalizing neighborhood $\mathcal{N}_a = \{(-1, 0), (0, 0), (1, 0), (2, 0)\}$.

(b) Cubic tile ($\Xi \equiv 0$), rule $R(X, Y) = 1 + XY + X^{-1}Y^{-1}$
$w(0, 0, 0) = w(0, 1, 1) = w(1, 0, 0) = w(1, 1, 1) = 1$, all other $w(r, i, j) = 0$.
Renormalizing neighborhood $\mathcal{N}_b = \{(-1, -1), (0, 0), (1, 1), (2, 2)\}$.

(c) Skew tile [$\beta_1 = 1$, all other Ξ-parameters are 0, see Fig. 2(b)]
Rule $R(X, Y) = X + XY + X^{-1}Y^{-1}$
Same weight-matrix as under (b)
Renormalizing neighborhood $\mathcal{N}_c = \{(0, -1), (1, 0), (2, 1), (3, 2)\}$.

(d) Skew tile ($\beta_1 = 2$, $\gamma_1 = 1$, all other Ξ-parameters are 0)
Rule $R(X, Y) = X + X^2 + X^{-1}Y^{-1}$
$w(0, 0, 0) = w(0, 1, 1) = w(1, 1, 0) = 1$, all other $w(r, i, j) = 0$.
Renormalizing neighborhood $\mathcal{N}_d = \{(0, 0), (2, 1), (4, 2)\}$.

(e) Skew tile [$\beta_1 = 1$, all other Ξ-parameters are 0, see Fig. 2(b)]
Rule $R(X, Y) = X^2 + Y^{-1}$
$w(0, 0, 1) = w(0, 1, 1) = w(1, 0, 0) = w(1, 1, 1) = 1$, all other $w(r, i, j) = 0$.
Renormalizing neighborhood $\mathcal{N}_e = \{(-1, 0), (2, 2)\}$.

(f) Cubic tile ($\Xi \equiv 0$), rule $R(X, Y) = X + Y + X^{-1} + XY + X^{-1}Y$
$w(0, 0, 0) = w(0, 1, 0) = w(1, 0, 1) = w(1, 1, 1) = 1$, all other $w(r, i, j) = 0$.
Renormalizing neighborhood $\mathcal{N}_f = \{(-1, 0), (0, 0), (1, 0), (2, 0), (-1, 1), (0, 1), (1, 1), (2, 1)\}$.

(g) Cubic tile ($\Xi \equiv 0$)
Rule $R(X, Y) = X + Y + XY + X^{-1}Y^{-1} + X^{-1}Y + XY^{-1}$
$W \equiv 1$.
Renormalizing neighborhood $\mathcal{N}_g = \{(-1, -1), (2, -1), (-1, 2), (2, 2)\}$.

(h) Cubic tile ($\Xi \equiv 0$)
Rule $R(X, Y) = 1 + X + Y$
$w(0, 1, 0) = w(0, 0, 1) = w(0, 1, 1) = w(1, 0, 0) = w(1, 0, 1) = 1$.
Renormalizing neighborhood $\mathcal{N}_h = \{(0, -1), (-1, 0), (1, 0), (-1, 1), (1, 1)\}$.

(i) Cubic tile ($\Xi \equiv 0$), rule $R(X, Y) = 1 + XY$
$w(0, 1, 0) = w(0, 1, 1) = w(1, 0, 1) = 1$, all other $w(r, i, j) = 0$.
Renormalizing neighborhood $\mathcal{N}_i = \{(-1, 0), (1, 0), (0, 1), (1, 1)\}$.

(j) Skew tile ($\beta_1 = 2$, $\gamma_1 = 1$, all other Ξ-parameters are 0)
Rule $R(X, Y) = 1 + X + X^{-2} + X^{-1}Y^{-1} + XY^{-1}$
$w(0, 0, 0) = w(0, 1, 1) = w(1, 1, 0) = 1$, all other $w(r, i, j) = 0$.
Renormalizing neighborhood $\mathcal{N}_j = \{(0, 0), (2, 1), (2, 2), (3, 1), (4, 2)\}$.

(k) Cubic tile ($\Xi \equiv 0$), rule $R(X, Y) = X + Y + XY$
$w(0, 0, 0) = w(0, 1, 0) = w(1, 0, 1) = w(1, 1, 1) = 1$, all other $w(r, i, j) = 0$.
Renormalizing neighborhood $\mathcal{N}_i = \{(-1, 0), (0, 0), (-1, 1), (0, 1)\}$.

Appendix B
The Quasiperiodic Seeds of Examples 5 and 6

Consider the one-dimensional bidirectional sequence $(s(k))_{k\in\mathbb{Z}}$ generated from a seed

$$[s(-1), s(0), s(1)] = [1, 1, 1]$$

through the left-propagating renormalizing equation ($\mathbb{Z}_2$ arithmetic!):

$$s(k) = s(k+1) + s(k+2) + s(k+3) + s\left(\frac{k}{2}\right) \quad \text{when} \quad k = \text{even}$$

$$s(k) = s(k+1) + s(k+2) + s(k+3) + s\left(\frac{k+1}{2}\right) \quad \text{when} \quad k = \text{odd}.$$

and through the right-propagating equation

$$s(k) = s(k-1) + s(k-2) + s(k-3) + s\left(\frac{k-2}{2}\right) \quad \text{when} \quad k = \text{even}$$

$$s(k) = s(k-1) + s(k-2) + s(k-3) + s\left(\frac{k-3}{2}\right) \quad \text{when} \quad k = \text{odd}\,.$$

This is the quasiperiodic initial configuration sequence for one of the CGI-problems in [Barbé *et al.*, 1995, Fig. 1(h)].

The quasiperiodic seed of Example 5 [generating the initial configuration of Fig. 16(c)] is defined as

$$(a_\lambda(k))_{k\in\mathbb{Z}} = (s(k))_{k\in\mathbb{Z}} \quad \text{for} \quad \lambda = -2,\, -1,\, 0,\, 1,\, 3$$

$$(a_\lambda(k))_{k\in\mathbb{Z}} = (s(k+1))_{k\in\mathbb{Z}} \quad \text{for} \quad \lambda = 2\,.$$

The quasiperiodic seed of Example 6 [generating the initial configuration of Fig. 16(d)] is taken to be

$$(a_\lambda(k))_{k\in\mathbb{Z}} = (s(k))_{k\in\mathbb{Z}} \quad \text{for} \quad \lambda = -2,\, -1,\, 0,\, 1$$

$$(a_\lambda(k))_{k\in\mathbb{Z}} = (s(k+1))_{k\in\mathbb{Z}} \quad \text{for} \quad \lambda = 2\,.$$