Solutions for certain number-conserving deterministic cellular automata

Janne V. Kujala* and Tuomas J. Lukka†
Department of Mathematical Information Technology, University of Jyväskylä, P.O. Box 35 (Agora), FIN-40351 Jyväskylä, Finland
(Received 28 April 2001; revised manuscript received 25 September 2001; published 18 January 2002)

We explain the unexpected behavior of the generalizations of cellular automation traffic models introduced in [H. Fukś and N. Boccara, Int. J. Mod. Phys. C **9**, 1 (1998)]. We analyze the steady-state flow in $\mathcal{R}_{m,k}$ as a function of the initial density; we show that these rules correspond to a system with an infinite number of different kinds of virtual particles interacting according to complex annihilation rules. From simple considerations, we are able to predict the unexpected cutoff of the average flow at unity observed by Fukś and Boccara. We present an efficient algorithm for determining the exact final flow from a given finite initial state. An analysis of this algorithm in the infinite limit using generating functions yields an exact polynomial equation between the flow and density for $\mathcal{R}_{m,k}$, of maximally $2(m+k)$th degree in both.

DOI: 10.1103/PhysRevE.65.026115 PACS number(s): 05.90.+m, 45.70.Vn, 89.40.+k, 02.60.−x

I. INTRODUCTION

The class of number-conserving cellular automata (NCAs) has received a great deal of attention [1–6]. In these cellular automata (CAs), the number (or sum) of nonzero sites remains the same over time, yielding effectively a system of interacting particles moving in a lattice. This is not only an appropriate model for many real-life phenomena but also allows results on particle kinetics to be applied to equivalent CAs [4].

The overall behavior of an NCA is described by a flow diagram, i.e., the flow of the particles as a function of their density. For many NCAs, the flow diagram is piecewise linear [2]. The best-known example is the binary one-dimensional deterministic CA rule number 184, in which each particle moves right to the next site if it is empty [1]. This rule is also known as the ''traffic rule'' because the particles can be interpreted as cars moving on a highway.

In Ref. [7], Fukś and Boccara proposed the deterministic rules $\mathcal{R}_{m,k}$ as a generalization of rule 184 (rule 184 is $\mathcal{R}_{1,1}$). When either $m=1$ or $k=1$, the flow diagram is piecewise linear, consisting of two pieces with slopes m and $-k$ (see Fig. 1). This pattern was expected to continue for both $m>1$ and $k>1$, but the numerical experiments in Ref. [7] yielded a surprise.

Contrary to the physical behavior and other, nondeterministic CA models of traffic [5,8–10], the peak of the flow diagram is cut off just below flow of one, creating a new phase between the two linear pieces. Fukś and Boccara [7] were perplexed by the qualitative change produced by their obvious-looking generalization and were unable to give an explanation for this behavior.

In this paper, we explain why the dynamics of the model produce the unexpected qualitative feature of the flow diagram and obtain an exact polynomial equation in the infinite limit. In the following sections, we first develop a formalism based on representing the road as a sequence of blocks (virtual particles) rather than single sites. We show that the average flow is fully determined by the number of these blocks in the steady state. In Sec. IV and V, we use this fact to obtain simple upper and lower limits for the average flow and present an efficient algorithm for calculating the steady-state flow from a given finite initial state. In Sec. VI, we consider the behavior of the algorithm in the infinite limit and derive a steady-state condition yielding an analytical solution in the case of an infinite space. Finally, in Sec. VII, we obtain a nontrivial upper limit for the expected average flow in a finite space.

II. GENERALIZED TRAFFIC RULES

The rules $\mathcal{R}_{m,k}$ are defined as follows [7]. A block of cars (ones) at most k units long moves right at most m units, or to the beginning of the next group. Thus, $\mathcal{R}_{1,1}$ corresponds to rule 184 and rules $\mathcal{R}_{m,1}$ and $\mathcal{R}_{1,k}$ generalize it by either increasing the speed limit or by allowing blocks of consecutive cars to move simultaneously. The same rule can also be expressed as follows: at each turn, each maximal match of 1^x0^y is replaced (see Fig. 2)

$$1^x0^y \rightarrow 1^{x-a}0^b1^a0^{y-b},$$

where $a=\min\{k,x\}$ and $b=\min\{m,y\}$. From this representation, the dualism between the motion of the cars under the rule $\mathcal{R}_{m,k}$ and the motion of the empty sites under rule $\mathcal{R}_{k,m}$ in the opposite direction, as mentioned in Ref. [7], is obvious.

The ''physical'' quantities of interest in systems that obey these rules are ρ, the density of ones, and ϕ, the flow, defined as $\rho\langle v\rangle$, where $\langle v\rangle$ is the average velocity of the cars. For finite-length systems, we write ϕ for the time-averaged steady-state flow from a single state and $\overline{\phi}$ for the average of ϕ over all states. For infinite-length systems, ϕ is the steady-state flow. The equation

$$\overline{\phi}_{\mathcal{R}_{m,k}}(\rho)=\overline{\phi}_{\mathcal{R}_{k,m}}(1-\rho) \tag{1}$$

expresses one consequence of the dualism discussed above. There are also other quantities such as acceleration, but these are beyond the scope of this paper.

*Email address: jvk@iki.fi
†Email address: lukka@iki.fi

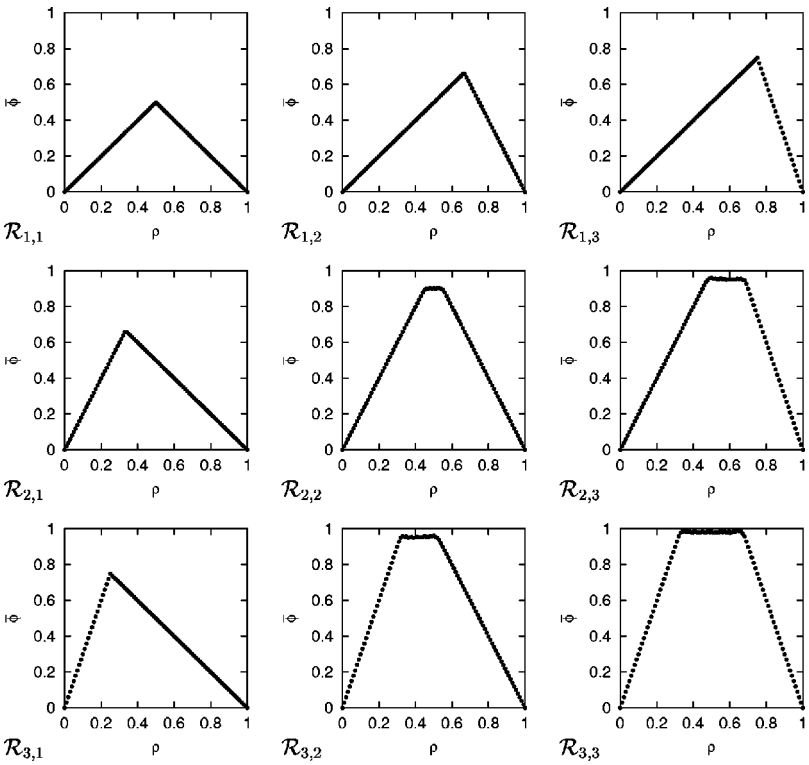

FIG. 1. Average flow over random initial configurations as a function of the density of cars ρ under the rule $\mathcal{R}_{m,k}$ for $m,k \in \{1,2,3\}$. The average flow never exceeds one regardless of the constraints m and k. Note that the diagram for rules $\mathcal{R}_{m,k}$ and $\mathcal{R}_{k,m}$ is symmetrical as shown by Eq. (1). $\mathcal{R}_{1,1}$ corresponds to rule 184. Each point is obtained as an average of 100 simulations on a lattice of size $L=1000$.

The dynamics of the class of NCAs with piecewise linear flow diagrams is considered in [4]. The relaxation to equilibrium is seen as ''defects'' propagating in opposite directions on a periodic background and annihilating upon collisions resembling ballistic annihilation [11]. For rule 184, the background is alternating zeros and ones and the defects are defined as two consecutive zeros or ones. When two defects collide, they annihilate completely into background. The steady-state flow is determined by the type and number of the remaining defects.

Rule $\mathcal{R}_{m,1}$ can be interpreted in a similar way, with the sequence $10^{m+x}1$ a type *A* defect for $x<0$, type *B* defect for $x>0$, and background for $x=0$ [4]. The defects may annihilate partially, but the final flow is still determined by the initial balance of the defects only.

Surprisingly, it turns out that $\mathcal{R}_{m,k}$ does not have a piecewise linear flow diagram for $m>1$ and $k>1$ (see Figs. 3 and 6). This is because the defects (however defined) interact nonlinearly, that is, their effects on the flow cannot be just summed up, and so the final flow depends on the initial order of the defects in addition to the balance between different types of defects. Because defects of a particular type do not always saturate the system in the middle phase, the final flow cannot be determined from just the density and so the average flow in the diagram depends on the expectation over different initial configurations. This expectation is not a linear function of the density; the flow diagram is curved in the intermediate phase.

Below, we show that rule $\mathcal{R}_{m,k}$ is equivalent to an interacting particle system with four indexed types of virtual particles similar to the defects of the above models. The general case is considerably more complex than $\mathcal{R}_{m,1}$ because the recursion resulting from the nonlinear annihilation rules has to be taken into account.

III. FUNDAMENTAL PROPERTIES OF $\mathcal{R}_{m,k}$

The flow of cars under rule $\mathcal{R}_{m,k}$ is easier to understand if the state of the road is considered as a sequence of blocks instead of single cars. As we shall see later, it is practical to distinguish between *short*, *just*, and *long* blocks, comparing the length of a block with *m* or *k* as follows: a block of zeros less than *m* sites long is a short block, more than *m* sites long is a long block, and exactly *m* sites long is a just block. For blocks of ones, the length is compared with *k* in a similar fashion. We say that a pair of a 0 block and a 1 block

road representation	symbol representation	groups
0100111100000000001000011011110111000001	$\star_{-2,-1}\ 1_{-1,2}\ 0_{7,-1}\ 0_{1,0}\ 1_{-2,2}\ 1_{-2,2}\ 0_{2,-1}$	7
1001110001100000000001001111011110001100 0	$1_{-1,1}\ \star_{0,0}\ 0_{7,-1}\ 1_{-1,2}\ 1_{-2,2}\ \star_{0,0}\ \star_{0,-1}$	7
0011000110001100000000011101111000011000 11	$\star_{-1,0}\ \star_{0,0}\ \star_{0,0}\ \diamond_{6,1}\ 1_{-2,2}\ \star_{0,0}\ \star_{0,0}$	7
1100011000110001100000001011110001100011000	$\star_{0,0}\ \star_{0,0}\ \star_{0,0}\ 0_{3,-1}\ 1_{-2,2}\ \star_{0,0}\ \star_{0,0}\ \star_{-1,0}$	8
0001100011000110001100001110001100011001 1	$\star_{0,0}\ \star_{0,0}\ \star_{0,0}\ \star_{0,0}\ \diamond_{1,1}\ \star_{0,0}\ \star_{0,0}\ \star_{-1,0}$	8
0110001100011000110001101000110001100110 0	$\star_{0,0}\ \star_{0,0}\ \star_{0,0}\ \star_{0,0}\ \star_{-2,-1}\ \star_{0,0}\ \star_{0,0}\ \star_{-1,0}\ \star_{0,0}$	9

FIG. 2. A sample time evolution under rule $\mathcal{R}_{3,2}$ from an initial state with $L=41$ into a cyclic state, expressed in both the road representation and the symbolic representation introduced in Sec. V in the text. Periodic boundary conditions are used.

TABLE I. The different types of states in the system. As discussed in the text, the existence of short and long blocks distinguishes the different types of states; just blocks are insignificant.

Lengths	$\rho_{0-}\geqslant 0,\rho_{0+}=0$	$\rho_{0-}>0,\rho_{0+}>0$	$\rho_{0-}=0,\rho_{0+}>0$
$\rho_{1-}\geqslant 0,\rho_{1+}=0$	Cyclic: Intermediate	Uncyclic	Cyclic: Free flowing
$\rho_{1-}>0,\rho_{1+}>0$	Uncyclic	Uncyclic ρ_G may increase	Uncyclic, ρ_G may increase
$\rho_{1-}=0,\rho_{1+}>0$	Cyclic: Congested	Uncyclic, ρ_G may increase	Uncyclic, ρ_G may increase

is a *group* and define ρ_G as the density of these groups. We also define ρ_{0+}, ρ_{0-}, and $\rho_{0=}$ as the densities of long, short, and just 0 blocks, respectively, and similar symbols for the 1 blocks.

The states of the system can be divided into nine different categories by the existence of short and long blocks, see Table I. In the following, we will consider the three cyclic types of states separately. To verify that these are the only cyclic states we first show that the following two cases are unstable: long and short blocks of one kind, $\rho_{1+}>0$ and $\rho_{1-}>0$, and long blocks of both kinds, $\rho_{1+}>0$ and $\rho_{0+}>0$. The other cases follow from the same proofs by duality.

First, we note that new long blocks can never form, because a nonlong block of cars moves continually to the right and, therefore, cannot absorb other cars from the left. This also implies that a nonlong block can only be absorbed to an already long block on the right. The duality proves the same for long 0 blocks. Furthermore, the length of a long block cannot grow, because long blocks emit just blocks whereas they can only absorb short or just blocks.

To show that the states that have both short and long blocks of one type are unstable, consider the sequence

$$1^x(0^{\leqslant m}1^k)^z0^{\leqslant m}1^y,$$

where $x<k$, $y>k$, and $0^{\leqslant m}$ represents a 0 block that is either short or just. On each of the z first steps, the block 1^y absorbs one just block and emits one just block from the other end—its length remains unchanged. However, on the $(z+1)$th step it absorbs a block of length x and emits a block of length k. Therefore, the number of short blocks has decreased by one, and the length of the long block has decreased by $k-x$, possibly transforming it into a just or short block. Mathematical induction using this argument shows that, if all 0 blocks are short or just, then during the simulation, the number of either short or long 1 blocks drops to zero. The number of groups does not change in this process. An important observation is that the long block behaves like a decaying quasiparticle that is moving in the opposite direction from the ones by continuously absorbing short or just blocks and emitting just blocks. Naturally, applying the dualism property proves the same for long and short 0 blocks.

Next, we show that if there are both long 1 blocks and long 0 blocks, the state is unstable. We can first apply the above property to show that either a long 1 block decays or it eventually meets a long 0 block moving in the opposite direction (or else there are no long 0 blocks left in the system). But when the long blocks meet, they react and annihilate each other partially or wholly: the group

$$0^x1^y, \tag{2}$$

where $x>m$ and $y>k$, emits just 0 blocks leftwards and just 1 blocks rightwards at each time step, reducing to

$$0^{x-m}1^{y-k}, \tag{3}$$

and increasing the number of groups by one. Therefore, eventually either the long 0 blocks, the long 1 blocks or both will be exhausted, which shows that the initial state is not cyclic.

This completes the study of the uncyclic states, showing that from any initial state we will finally end up in one of the remaining states shown in Table II. We discuss these three cyclic types of states separately below.

It is fairly easy to see that if there are only short and just blocks, then the 1 blocks move in the positive direction and the 0 blocks move in the negative direction but the number of blocks and the distribution and relative order of the 0 and 1 blocks among themselves do not change—the state is obviously cyclic. To evaluate the flow, we first note that in such a state each block of cars travels on each step on average $(1-\rho)/\rho_G$ units forwards, which, when multiplied by the density of cars yields

$$\phi=\frac{\rho(1-\rho)}{\rho_G}, \quad \rho_{0+}=0, \ \rho_{1+}=0. \tag{4}$$

As this class of states does not correspond to either the free-flowing or the congested phase of the simpler traffic rules $\mathcal{R}_{1,k}$ and $\mathcal{R}_{m,1}$, we term it, for want of a better name, intermediate.

The free-flowing states where all 0 blocks are long or just and all 1 blocks are short or just are also simple. All the cars obviously move forwards at maximum speed and consequently, these states are also cyclic, with

$$\phi=m\rho, \quad \rho_{0-}=0, \ \rho_{1+}=0. \tag{5}$$

Applying the dualism between zeros and ones we obtain the formula

TABLE II. The three different types of cyclic states for $\mathcal{R}_{m,k}$ and the formula for the flow ϕ in each.

Description	Conditions	ϕ
Free flowing	$\rho_{1+}=0, \quad \rho_{0-}=0$	mp
Intermediate	$\rho_{+}=0, \quad \rho_{0+}=0$	$\rho(1-\rho)/\rho_G$
Congested	$\rho_{1-}=0, \quad \rho_{0+}=0$	$k(1-\rho)$

$$\phi = k(1-\rho), \quad \rho_{0+} = 0, \quad \rho_{1-} = 0, \tag{6}$$

for the opposite case: congested states with long or just 1 blocks and short or just 0 blocks.

We can summarize the above by noting that ρ and the final ρ_G determine the type of the cyclic state. This is because the final average lengths of 0 and 1 blocks can hold for only one type of a cyclic state: in the intermediate phase blocks must on the average be short or just whereas in the free flowing and congested phases either 0 or 1 blocks must be long and the other blocks short or just as shown in Table II. Writing these conditions in terms of ρ and ρ_G allows us to combine the above evaluations of ϕ into one surprisingly simple formula.

Proposition 1. The average flow over a cycle in any cyclic state of $\mathcal{R}_{m,k}$ *is*

$$\phi = \min\{m\rho, \rho(1-\rho)/\rho_G, k(1-\rho)\}.$$

The treatment of long blocks above also yields the following proposition.

Proposition 2. During evolution of the system, ρ_G can never decrease. It can increase only when $\rho_{0+} > 0$ and $\rho_{1+} > 0$.

These two propositions give the system its interesting characteristics: the final flow is completely determined by the final value of ρ_G, which in turn depends on the intricate reactions of the long blocks.

As a direct consequence of these propositions, it is straightforward to obtain crude upper and lower limits under both finite and infinite length (see Fig. 3).

Proposition 3. The flow of any cyclic state (and thus the average flow over different states) satisfies

$$\phi \geqslant \min\{m\rho, |\rho - 1/2| + 1/2, k(1-\rho)\}, \tag{7}$$

$$\phi \leqslant \min\{m\rho, k(1-\rho)\}, \tag{8}$$

for any $0 \leqslant \rho \leqslant 1$ and any size lattice.

The lower limit is obtained through considering what is the greatest possible number of groups. For a lattice of size L, this is easily seen to be

$$N_{G,\,\max} = L\left(\frac{1}{2} - \left|\rho - \frac{1}{2}\right|\right),$$

which together with Proposition 1 yields the first part. The second inequality follows directly from Proposition 1.

When $m=1$ or $k=1$, this proposition reduces to the well-known flow formula

$$\phi = \min\{m\rho, k(1-\rho)\},$$

discussed, e.g., in Ref. [7].

IV. SIMPLE UPPER AND LOWER LIMITS AT INFINITE LENGTH

At infinite length, we can use block probabilities

$$p(a_1\, a_2 \ldots a_n) = \rho^{\Sigma_i [a_i = 1]} (1-\rho)^{\Sigma_i |a_i = 0|}, \tag{9}$$

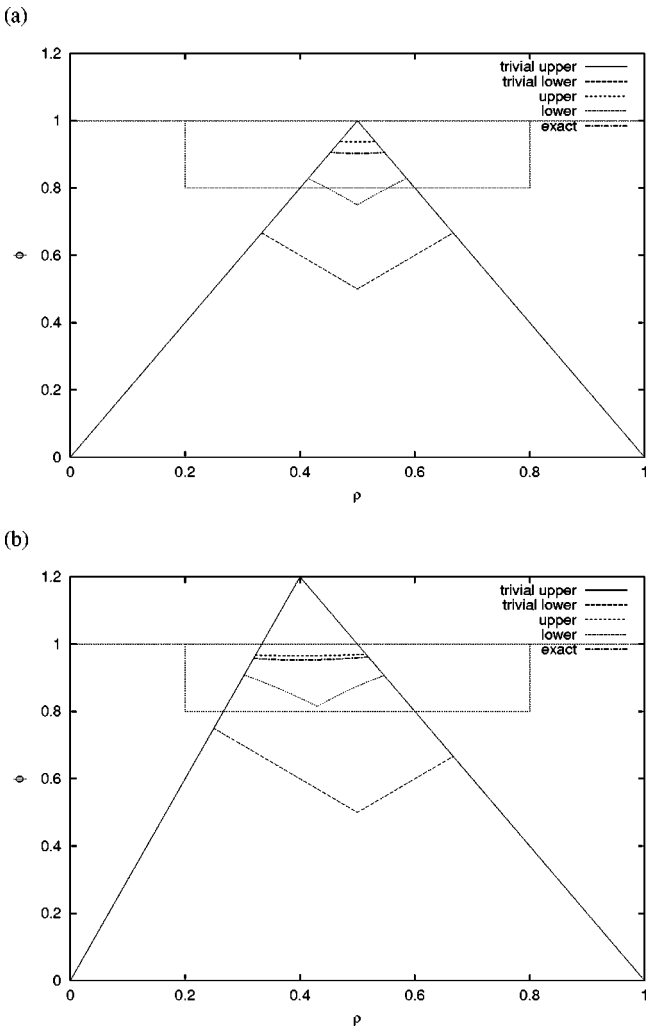

FIG. 3. Different limits and the exact solution for the flow ϕ as a function of density ρ at infinite length for (a) $\mathcal{R}_{2,2}$ and (b) $\mathcal{R}_{3,2}$. The trivial limits are given by Eqs. (7) and (8). Equations (11) and (12) yield the tighter limits. The enclosed regions are shown magnified in Fig. (6).

in the initial random state in order to calculate various statistics. The brackets $[\cdots]$ represent Iverson's notation (p. 11 of Ref. [12]), evaluating to 1 if the enclosed statement is true and to 0 otherwise.

The frequency of groups in the initial, random state is obviously given by the density of group edges

$$\rho_{G,\,\text{initial}} = p(01) = \rho(1-\rho)\,.$$

Proposition 2 tells us that

$$\rho_G \geqslant \rho_{G,\,\text{initial}}\,.$$

Combining these with Proposition 1 yields the important upper limit

$$\phi \leqslant \min\{m\rho, 1, k(1-\rho)\}, \tag{10}$$

which is valid for *any* $\mathcal{R}_{m,k}$, as Fukś and Boccara observed experimentally in Ref. [7]. The expected flow is cut off be-

cause the initial groups can never combine into fewer groups, which limits the maximum speed of cars. We shall show in Sec. VII that this limit is also valid for finite-length systems.

This limit does not take into account the dynamics of the system but assumes that the final state has the same number of groups as the initial state. It is possible to improve this upper limit by explicitly including some cases where certain initial configurations are known to produce new groups. For adjacent long blocks of zeroes and ones that directly react with each other as shown by Eqs. (2) and (3), we have

$$\begin{aligned}\rho_G &\geqslant \sum_{x,y\geqslant 1} \min\left\{\left[\frac{x}{m}\right],\left[\frac{y}{k}\right]\right\} p(10^x1^y0)\\ &= \sum_{r\geqslant 0} p(0^{mr+1}1^{kr+1})\\ &= \frac{\rho(1-\rho)}{1-\rho^k(1-\rho)^m},\end{aligned}$$

which yields the tighter upper limit

$$\phi \leqslant \min\{m\rho, 1-\rho^k(1-\rho)^m, k(1-\rho)\}, \tag{11}$$

when combined with Proposition 1.

A lower limit can be estimated by computing the density of groups that could arise by having all long blocks split maximally. For ones, this is

$$\rho_G \leqslant \sum_{x\geqslant 1}\left[\frac{x}{k}\right] p(01^x0) = \sum_{r\geqslant 0} p(01^{kr+1}) = \frac{\rho(1-\rho)}{1-\rho^k}.$$

Combining with the same formula for zeroes yields

$$\phi \geqslant \min\{m\rho, \max\{1-\rho^k, 1-(1-\rho)^m\}, k(1-\rho)\}. \tag{12}$$

Figure 3 shows the accuracy of these limits for $\mathcal{R}_{2,2}$ and $\mathcal{R}_{3,2}$. For larger m and k, the limits become tighter, converging to unity geometrically as m and k tend to infinity.

V. EFFICIENT ALGORITHM FOR COMPUTING THE STEADY-STATE FLOW

It is not necessary to carry out the simulation to find the number (or density) of groups in the final state, and thus the steady-state flow. In this section, we present an $O(L)$ algorithm for finding the final number of groups from a given starting state with periodic boundary conditions in a lattice of length L. This is interesting from two different perspectives: first, it makes it possible to calculate the final flow for long strings more effectively. Additionally, it helps us to understand the dynamics of the system and derive analytic results on the behavior of the system in the next sections.

Let us define the symbols $\star_{a,b}$, $\diamondsuit_{a,b}$, $0_{a,b}$, and $1_{a,b}$ all to correspond to $0^{a+k}1^{b+m}$ for different a and b,

$$0^{a+k}1^{b+m} = \begin{cases} \star_{a,b}, & a\leqslant 0 \quad \text{and} \quad b\geqslant 0,\\ 0_{a,b}, & a>0 \quad \text{and} \quad b\leqslant 0,\\ 1_{a,b}, & a\leqslant 0 \quad \text{and} \quad b>0,\\ \diamondsuit_{a,b}, & a>0 \quad \text{and} \quad b>0. \end{cases} \tag{13}$$

We represent the initial state in this notation and then carry out a series of string replacements. The final number of groups is obtained as the initial number of symbols plus the number of extra groups created by the replacements as shown below.

Diamonds react as follows:

$$\diamondsuit_{a,b} \rightarrow ?_{a-m,b-k}, \quad \Delta g = +1, \tag{14}$$

where the wildcard symbol ? represents the correct symbol for the new indices from Eq. (13) and Δg represents the number of new groups created by the reaction.

Stars can be collected, along with zeroes and ones, but zeroes only from the right and ones only from the left. No new groups are created.

$$\star_{a,b}\star_{c,d} \rightarrow \star_{a+c,b+d}, \quad \Delta g = 0, \tag{15}$$

$$0_{a,b}\star_{c,d} \rightarrow ?_{a+c,b+d}, \quad \Delta g = 0, \tag{16}$$

$$\star_{a,b}1_{c,d} \rightarrow ?_{a+c,b+d}, \quad \Delta g = 0. \tag{17}$$

Finally, when a 0 and a 1 meet, it is possible to form new groups,

$$0_{a,b}1_{c,d} \rightarrow ?_{a+c,b+d}, \quad \Delta g = 0. \tag{18}$$

Note that new groups are not created directly by this reaction, but the result may be a $\diamondsuit$.

The above replacements recursively evaluate the reactions between long blocks. To prove the correctness of a maximal application of the replacements, we consider the reactions occurring in an actual simulation. As described in Sec. III, all new groups arise from the annihilations of long blocks and the length of a long block can never grow. Thus, the number of extra groups is dictated by the sequence of short blocks that reacting long blocks must absorb before annihilating. In the following, we shall show that each of the nested annihilations in an actual simulation is correctly represented by the string replacements.

First, consider a subsequence starting with a long 0 block and ending with a long 1 block with only short and just blocks between them. Suppose further that the long blocks are long enough to actually meet before turning into short of just blocks. Then a maximal application of Eqs. (15)–(18) to the subsequence subtracts the total ''shortness'' of the short blocks from the long blocks yielding a diamond symbol, which correctly represents what remains of the long blocks when they meet. The diamond will then annihilate according to Eq. (14) and yield the correct number of extra groups and a symbol representing the remainder (either a $\star$, if both long blocks are completely annihilated, or a 0 or 1, if the annihilation is partial). This remainder symbol is exactly what would be the result of an actual simulation of the subsequence minus the just blocks that the sequence would have emitted from both ends. Because the just blocks do not affect long blocks outside the subsequence, they can be disregarded.

Suppose then that either long block would decay before meeting the other long block. Then an application of Eqs.

0. Initialize $g \leftarrow 0$ and create an empty stack.

1. If there are no symbols left in the input string, go to step 6.

2. (*Input new symbol*) Read the next symbol and push it to the stack, and set $g \leftarrow g+1$

3. (*Reaction to create new groups*) While the top symbol is a $\diamond$, i.e. (a,b) with $a>0$ and $b>0$, subtract (m,k) from the symbol, and set $g \leftarrow g+1$.

4. (*Drop reaction result down stack*) If the top symbol is a non-0 and the symbol below it is a 0, pop the two symbols off the stack and push their sum and go to step 3.

5. Go back to step 1.

6. (*Finish up by wrapping the stack*) If the top symbol is a non-0 and the bottom symbol of the stack is a 0, remove the bottom symbol and push it back to the top of the stack, and continue from step 3.

7. Terminate. The variable g now contains the total number of groups that this string will have in the steady state.

FIG. 4. This algorithm calculates the final number of groups in the given symbol string with periodic boundary conditions. The symbols $?_{a,b}$ are represented as pairs (a,b) and the input string is read left to right. Addition between pairs is defined as $(a,b)+(c,d)=(a+c,b+d)$. Step 3 corresponds to Eq. (14) and steps 4 and 6 both corresponds to Eqs. (16) and (18) in the text.

(16)–(18) to the subsequence will at some point turn either the 0 or 1 symbol into a $\star$ (or the whole sequence can turn into a 0, 1, or $\star$). A long block at one or both ends of the subsequence has thus turned into a just or short block, making it possible for outside long blocks to react over the remaining just and short blocks of the subsequence. Again, the just blocks that a decaying long block would have emitted are disregarded as they do not affect the length or order of other long blocks.

These are in fact all the cases that need to be considered, as the above cases can be applied recursively to the results of inner annihilations. If neither case applies, we have found the final number of groups, because either long 0 blocks or long 1 blocks have then been exhausted and so there cannot be any other reactions in an simulation nor can the replacements form any new $\diamond$'s. On the other hand, each of the applicable reactions will be carried out at some point of a maximal application of the replacements, because the 0 and 1 symbols at the ends or a $\diamond$ do not interact with outside symbols. The replacements may evaluate the decay of the annihilating long blocks in different order from the actual simulation, but as the long blocks must absorb all short blocks between them before they can meet and the short blocks cannot disappear unless absorbed by a long block, the result is still the same.

Note that the remainder of a $\diamond$ may be any non-$\diamond$ symbol, causing complicated recursion as the result of an annihilation affects further replacements. For example, consider the replacements

$$\underbrace{\underbrace{0\star\underbrace{0\star1}_{1}\star\underbrace{0\star1}_{\star}}_{0}1}_{1}1$$

where the innermost reactions must be evaluated before the outer reactions can be considered regardless of the order of the possible replacements.

The iteration of Eqs. (14)–(18) can be carried out by the stack algorithm shown in Fig. 4. Each symbol of the input string requires $O(l)$ operations to be dealt with in this algorithm, where l is the bit length of the symbol. This also includes the final steps to wrap the stack. Therefore, the running time of this algorithm is obviously $O(L)$ for a lattice of length L.

The worst-case time for simply running the cellular automaton simulation, on the other hand, is $\Omega(L^2)$ since in this system faraway cars do interact with each other. For ex-

time	groups	dropped symbols	active stack	remaining input
0	0			$\star_{-2,-1}\ 1_{-1,2}\ 0_{7,-1}\ 0_{1,0}\ 1_{-2,2}\ 1_{-2,2}\ 0_{2,-1}$
1	1		$\star_{-2,-1}$	$1_{-1,2}\ 0_{7,-1}\ 0_{1,0}\ 1_{-2,2}\ 1_{-2,2}\ 0_{2,-1}$
2	2	$\star_{-2,-1}$	$1_{-1,2}$	$0_{7,-1}\ 0_{1,0}\ 1_{-2,2}\ 1_{-2,2}\ 0_{2,-1}$
3	3	$\star_{-2,-1}\ 1_{-1,2}$	$0_{7,-1}$	$0_{1,0}\ 1_{-2,2}\ 1_{-2,2}\ 0_{2,-1}$
4	4	$\star_{-2,-1}\ 1_{-1,2}$	$0_{7,-1}$ $0_{1,0}$	$1_{-2,2}\ 1_{-2,2}\ 0_{2,-1}$
5	5	$\star_{-2,-1}\ 1_{-1,2}$	$0_{7,-1}$ $0_{1,0}$ $1_{-2,2}$	$1_{-2,2}\ 0_{2,-1}$
	5	$\star_{-2,-1}\ 1_{-1,2}$	$0_{7,-1}$ $1_{-1,2}$	$1_{-2,2}\ 0_{2,-1}$
	5	$\star_{-2,-1}\ 1_{-1,2}$	$\diamond_{6,1}$	$1_{-2,2}\ 0_{2,-1}$
	6	$\star_{-2,-1}\ 1_{-1,2}$	$0_{3,-1}$	$1_{-2,2}\ 0_{2,-1}$
6	7	$\star_{-2,-1}\ 1_{-1,2}$	$0_{3,-1}$ $1_{-2,2}$	$0_{2,-1}$
	7	$\star_{-2,-1}\ 1_{-1,2}$	$\diamond_{1,1}$	$0_{2,-1}$
	8	$\star_{-2,-1}\ 1_{-1,2}$	$\star_{-2,-1}$	$0_{2,-1}$
7	9	$\star_{-2,-1}\ 1_{-1,2}\ \star_{-2,-1}$	$0_{2,-1}$	
	9	$1_{-1,2}\ \star_{-2,-1}$	$0_{2,-1}$ $\star_{-2,-1}$	
	9	$1_{-1,2}\ \star_{-2,-1}$	$\star_{0,-2}$	

FIG. 5. Evolution of the stack of the algorithm in Fig. 4 operating on the sample string in Fig. 2. In the end, the number of extra groups is the same as in Fig. 2. The doubled line signifies the end of the input string, after which the stack is wrapped over (cf. step 6 of the algorithm in the Fig. 4). The division to dropped symbols and active stack is implicit—it is only significant for the Markov model analysis of the algorithm in Appendix A. The boxes depict the "lifetimes" of the substacks started by each input symbol. The time steps of the Markov model are at the bottom lines of these boxes, where the substacks are finished.

ample, the initial state $0^{m+1}(1^k0^m)^{T-1}1^{k+1}$ (in the road representation) would require T simulation steps to reach a cyclic state.

Figure 5 illustrates an application of the algorithm on an example string.

VI. ANALYTICAL SOLUTIONS AT THE INFINITE LIMIT

Infinite limits of systems are often easier to solve than finite system. This is also the case with $\mathcal{R}_{m,k}$. In the Appendixes, we examine the behavior of the stack machine algorithm as the length approaches infinity. In Appendix A, we regard the evolution of the stack configurations as a Markov process and derive a set of equations for the stationarity of the probabilities of stack configurations. In Appendix B, we solve these equations to determine the probabilities of different annihilations and thus the steady-state flow.

The most important results of the Appendixes are summarized below:

Proposition 4. The flow at infinite time for random infinite strings in the intermediate phase is C, where $1/C$ is the expected number of groups arising per each input symbol and is determined by Eqs. (B9)–(B12). The flow depends on ρ only through the quantity $A=(1-\rho)^m\rho^k$.

For any given m and k, Eqs. (B9)–(B12) for A and C can be expanded to a polynomial equation, which is easily seen to be second degree in A and at most $2(m+k)$th degree in C. In the special case of $m=k$, we can solve ρ from A in closed form and obtain the density as a function of flow. In practice the degree of C seems to reduce to $m+k+1$, and we conjecture that this holds for all m and k. For example, in the case of $m=k=2$, the equation can be written out as

$$16A^2+8AC^2-36AC^3+(1+27A)C^4-C^5=0.$$

Proposition 5. For any given m and k, the flow ρ at infinite time and the density ρ in the intermediate phase can be related by a polynomial equation maximally of degree $2(m+k)$.

Furthermore, the phase transitions occur where C as a function of ρ crosses $m\rho$ or $k(1-\rho)$, the flow of free flowing and congested phases. For example, when $m=k=2$ the phase transitions can be solved to be exactly at

$$\rho=\frac{1}{2}\pm\frac{1}{7}\left(2\sqrt{2}-\frac{5}{2}\right).$$

VII. UPPER LIMIT FOR STEADY-STATE FLOW IN FINITE SYSTEMS

Carrying out calculations for finite systems is considerably more difficult, since the probabilities are no longer independent of each other. However, the following interesting limit can be derived.

Proposition 6. The average flow in steady states starting from random binary strings of length L with $0<\rho<1$ satisfies

$$\overline{\phi}\leqslant\min\left[m\rho,1-\binom{L}{\rho L}^{-1},k(1-\rho)\right],$$

where equality applies at least when $k\geqslant L-1$ and $m\geqslant L-1$.

Proof. If $0<\rho<1$, the number of different initial states with a given number of groups can be counted by considering different ways of placing the group boundaries on the string. The distribution of N_G for random binary strings can be simplified to

$$p(N_G)=\begin{cases}\dfrac{L}{N_G}\dfrac{\binom{\rho L-1}{N_G-1}\binom{(1-\rho)L-1}{N_G-1}}{\binom{L}{\rho L}} & \text{if} \quad N_G>0\\ 0 \quad \text{if} \quad N_G\leqslant 0.\end{cases}$$

Using basic binomial coefficient sum formulas (see, e.g., Ref. [13]) and noting that N_G cannot be zero, the expected value of L/N_G is

$$\left\langle\frac{L}{N_G}\right\rangle=\sum_{N_G}N_Gp(N_G)=\frac{1}{\rho(1-\rho)}\left[1-\binom{L}{\rho L}^{-1}\right].$$

From this, the formula in the proposition follows. Finally, the equality follows simply from the fact that if the condition given holds, no groups can split.

Note that with reasonable L and ρL, the reciprocal of the binomial coefficient is negligibly small compared to 1.

VIII. SIMULATIONS

Simulations were carried out to test the theoretical results. For small L, complete summations were possible so the simulated curves are, in fact, exact. For large L, a number of random initial configurations were generated and the evolution of the system simulated. Since the resulting steady-state flow under $\mathcal{R}_{m,k}$, when $m>1$ and $k>1$, depends on the whole initial state (and not just ρ, as for when $m=1$ or $k=1$), the samples so simulated will in general not have the same flow. Therefore, the standard deviation is displayed along with the average of the resulting flows, giving an idea of the strength of the fluctuations. As L tends to infinity, the fluctuations slowly average out, displaying the usual $1/\sqrt{L}$ behavior for the standard deviation.

Figure 6 depicts the simulated flow and exact solution for infinite space. The theoretical solution agrees well with simulated results. Figure 7 shows the simulated flow and upper and lower limits for finite space. For m, $k\geqslant L-1$ the upper limit is exact as confirmed by the simulation.

IX. CONCLUSIONS

In this paper, we have solved the behavior of the generalized traffic rules $\mathcal{R}_{m,k}$ for infinite lengths of road and uniform random initial conditions.

We have derived an efficient algorithm for computing the average flow from an initial state under the generalized traffic rules $\mathcal{R}_{m,k}$. The idea behind the algorithm is an appropriate representation of the road as a string of 0^x1^y blocks instead of single sites. Finding the average flow can be reduced to finding the number of these blocks in the cyclic state.

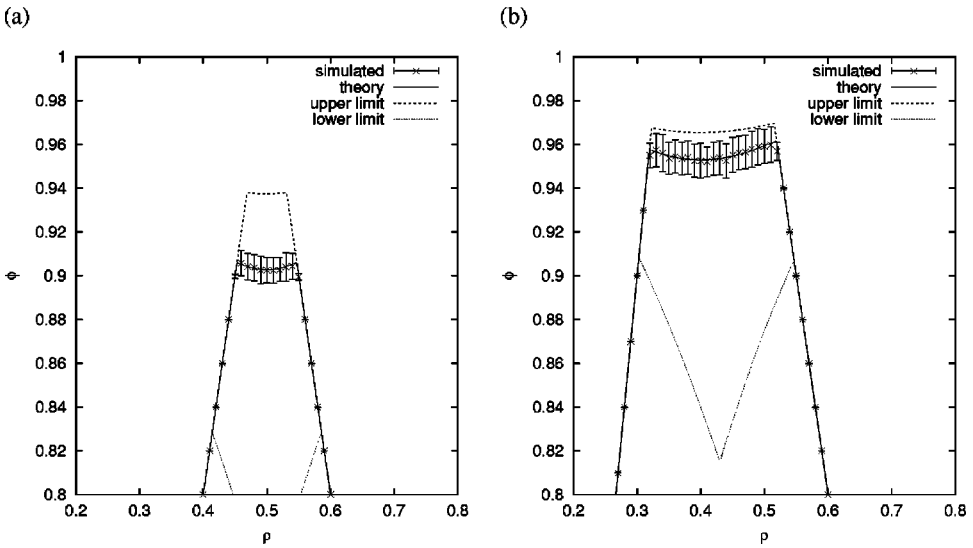

FIG. 6. Infinite-space behavior of flow as a function of density under (a) $\mathcal{R}_{2,2}$ and (b) $\mathcal{R}_{3,2}$. Average of 100 simulations with $L=10\,000$, theoretical flow, and upper and lower limits given by Eqs. (11) and (12).

The algorithm works by decoupling the time from the simulation and considering directly the different reactions that would occur during the evolution of the system. Analysis of the algorithm in the infinite limit yields an exact solution for the flow in an infinite space. Simulated results agree perfectly with the analytic solution.

Finite-space behavior is more complex because single sites are no longer independent. We have, however, been able to obtain for the average flow a nontrivial upper limit, which is exact for m, $k \geqslant L-1$.

ACKNOWLEDGMENTS

The authors would like to thank Antti-Juhani Kaijanaho and Rauli Ruohonen for discussions and comments on this manuscript.

APPENDIX A: STEADY-STATE CONDITIONS FOR THE STACK MACHINE AT THE INFINITE LIMIT

Consider the symbols and reactions defined above in Sec. V. All new groups are created from the diamonds, which in turn can only arise when a 0 symbol is combined with a following 1 symbol. The algorithm in Fig. 5 uses this fact to find the final number of groups by only tracking the reactions of 0 symbols. It scans through the input string linearly, from left to right, maintaining a stack of the processed symbols with all reactions of 0's and $\diamondsuit$'s carried out. This means that all the remaining 0 symbols end up on top, which we shall now call the *active part* of the stack. When a non-0 input symbol consumes all the 0's we say that the symbol is dropped off from the bottom of the active stack into the *inactive part*, as the symbol can no longer create new groups with the *following* symbols.

In finite systems with periodic boundary conditions, the processing of the input string is divided into two parts. First, the input string is scanned as above. When all of the input has been read, the inactive part of the stack, comprising of 1's and $\star$'s, is reprocessed with the zeroes in the active part, since the 1's on the bottom of the stack may react with the 0's on top, producing new groups. The relative effect of this

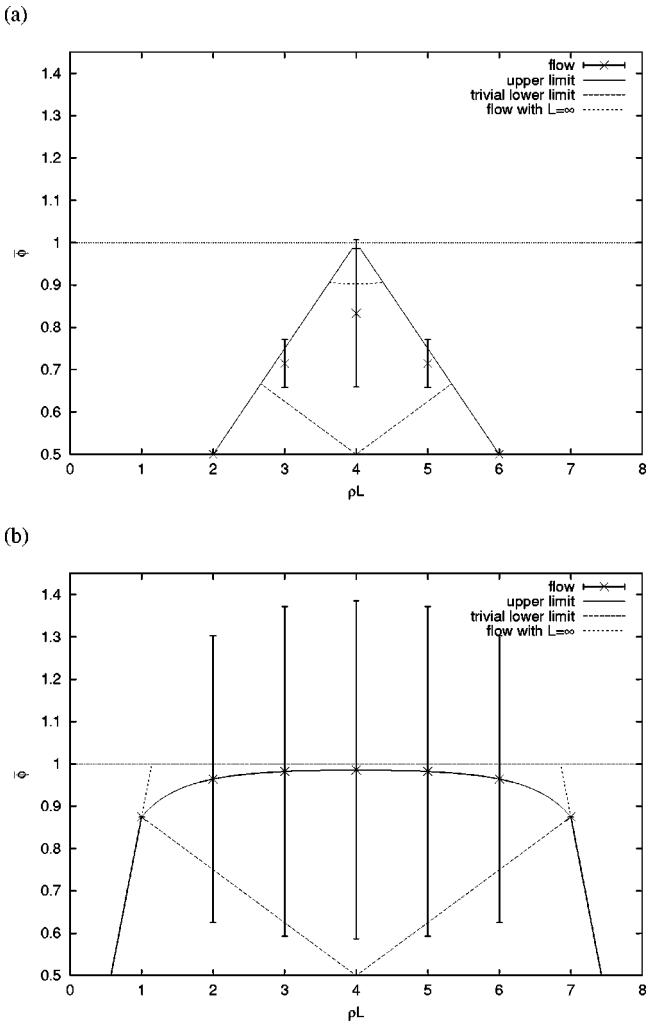

FIG. 7. Finite-space behavior of the average flow $\overline{\phi}$ as a function of density under (a) $\mathcal{R}_{2,2}$ and (b) $\mathcal{R}_{7,7}$ when $L=8$. All points represent either evaluated formulas or averages calculated over all configurations. The trivial lower limit is given by Eq. (7). The upper limit given by Proposition 6 is exact for m, $k \geqslant L-1$ as demonstrated by (b).

wrapping diminishes as the length approaches infinity: it is easily shown that we can ignore all symbols dropped to the inactive part in the infinite limit.

To obtain the limiting flow, we thus need to evaluate the average number of new groups produced as the stack algorithm processes a new symbol. When new symbols are input, the active stack can either grow infinitely or remain finite. If the active stack remains finite, the probabilities of different active stacks will eventually reach a stationary distribution. Once the stationary distribution is known, it is straightforward to calculate the expected number of new groups for a random input symbol.

If, on the other hand, there are not enough 1's and $\star$'s to annihilate the 0's and the stack grows infinitely, we can use the dualism property and consider the thus finite stack of 1's instead of 0's. It turns out later that we do not even have to consider this dualism explicitly, as the symmetry of the equations is restored in Appendix B.

Formally, we regard the evolution of the active stack as a Markov process. The Markov property is satisfied as the next state depends only on the current state and the upcoming independently distributed random symbol. Clearly, the process can reach each possible stack configuration from every state and has a positive probability to remain in its current state (the symbol $\star_{0,0}$). Furthermore, given that the active stack will not grow infinitely, the process will return to every state an infinite number of times with probability one. This means that the Markov chain is irreducible, aperiodic, and Harris recurrent and therefore will converge to a unique stationary distribution (see, e.g., [14] or Proposition 6.3 of Ref. [15]).

An essential property is that the algorithm can be applied independent of the lower levels for each stacked symbol until that substack is finished, that is, until the lowest level of the substack turns into a 1 or a $\star$, which happens either immediately, if the stacked symbol is already a 1 or a $\star$, or when a 1 or a$\star$ on a higher level falls to the bottom level consuming all 0's on the substack. This allows us to consider each level of the active stack as the bottom of an identically distributed substack. The distribution is particularly interesting when a substack is just finished. At these times the active stack consists of zeroes at the bottom of each substack and of the 1 or $\star$ that finishes the topmost substack (see Fig. 5).

Suppose $p(?_{x,y})$ is the distribution of symbols seen on the bottom level of the active stack at each time a substack is finished. Then, if the symbol is a 0, the same distribution of symbols will be seen on the bottom of the substack above that symbol. Thus, a symbol distribution defines a distribution for stack configurations. Note that an arbitrary stack distribution cannot be represented by such a symbol distribution but it is required that the stack symbols are identically and independently distributed and that the height of the stack is implied by the stack symbols as described above. Furthermore, even though each input symbol starts a substack so that there are the same number of time steps as there are input symbols, the substacks are finished out-of-synch with the time steps of the algorithm. This complication, however, is inconsequential as the expected density of new groups on each time step is still the same as the expected per symbol density.

For simplicity, we consider the input symbol distribution $p_i(?_{x,y})$ to represent what remains of the symbol after initial reactions of $\diamondsuit$ symbols. This difference is only conceptual as the algorithm would immediately carry out the initial reactions for each input symbol anyway. It is easy to see that the resulting distribution must still be geometric; only a constant is required to normalize the lack of $\diamondsuit$'s. The normalized input distribution is defined for indices in the set

$$D := \{(x,y): x > -m,\ y > -k,\ [x>0][y>0]=0\}$$

as

$$p_i(?_{x,y}) := = \frac{A(1-\rho)^x \rho^y}{1-A}, \tag{A1}$$

where we define the symbol A to represent the quantity

$$A := (1-\rho)^m \rho^k, \tag{A2}$$

which occurs often in the formulas below. This quantity is the initial probability of a $\diamondsuit$, and the initial reactions produce a density of $A/(1-A)$ new groups (cf. Sec. IV). Here and in the following, we use densities relative to the initial symbol density 1 in the string of symbols.

In our model the initial stack distribution is defined by $p(\cdot\,|t=0) := p_i(\cdot)$ corresponding to the distribution of stacks that results from running the algorithm on random input until the first non-0 symbol is stacked in, i.e., the first time a substack is finished (cf. Fig. 5).

The transition from a symbol distribution $p(\cdot\,|t)$ to the distribution $p(\cdot\,|t+1)$ on next time step can be defined by considering the possible ways for a given symbol to arise on each level of the stack: A symbol can be the result of a 0 and a non-0 above it reacting as per Eqs. (16) and (18). The reaction will result in a non-$\diamondsuit$ symbol $?_{x,y}$ with probability

$$p_c(?_{x,y}|t) = \sum_{a<x;\, a\leqslant 0} \sum_{b\leqslant 0} p(0_{x-a,b}|t)\, p(?_{a,y-b}|t).$$

If, on the other hand, the result of the reaction is a diamond, it will further react, possibly several times, according to Eq. (14) and yield $?_{x,y}$, where $(x,y)\in D$, with probability

$$p_d(?_{x,y}|t) = \sum_{r\geqslant 1} \sum_{a\leqslant 0} \sum_{b\leqslant 0} p(0_{x-a+rm,b}|t)\, p(1_{a,y-b+rk}|t).$$

In case there is another 0 on top of a $0_{x,y}$, no reactions will occur and the $0_{x,y}$ remains for the next time step. Finally, if the symbol is not a 0, it falls off from the bottom of the substack and is replaced by a fresh symbol from the input distribution. Thus, the transition function can be written as

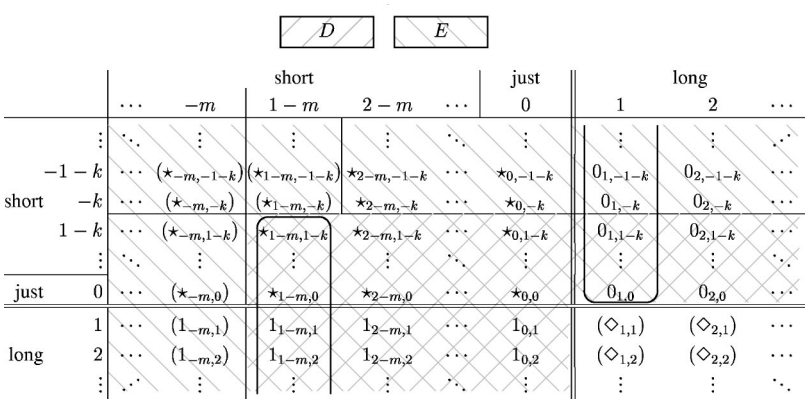

FIG. 8. The possible index combinations for different symbols in the Markov model for the algorithm discussed in Appendix A. By considering the possible outcomes of combined 0 and non-0 symbols, it can be seen that the parenthesized symbols cannot occur. The sets D and E defined in Appendix A are depicted by the different shadings. The marked sequences are represented by the generating functions $F_{1-m}(z)$ and $F_1(z)$ discussed in Appendix B.

$$p(?_{x,y}|t+1)=[(x,y)\in D]p_d(?_{x,y}|t)+[(x,y)\in E]p_c(?_{x,y}|t)+[x>0]p(?_{x,y}|t)p(0|t)+(1-p(0|t))p_i(?_{x,y}). \tag{A3}$$

where $p(0|t)$ denotes the total probability of 0 symbols and the set E is defined as the complement of diamonds,

$$E:=\{(x,y):[x>0][y>0]=0\}.$$

Note that the transition function does not define a Markov chain for symbols but it implicitly defines a linear Markov operator for the subspace of stack distributions that are determined by symbol distributions.

The stack distribution defined by a symbol distribution is stationary when

$$p(?_{x,y}|t+1)=p(?_{x,y}|t). \tag{A4}$$

Thus, we need to find a symbol distribution $p(?_{x,y})$ corresponding to the unique stationary stack distribution. The symbol distribution can then be used to determine probabilities for different reactions.

Figure 8 represents the possible indices for different stack symbols. It is easy to see why some symbols cannot occur at all, but even more can be said. The distribution of the stack symbols retains some of the geometric properties of the input distribution. The distribution of 0 symbols is geometric in its x index and the distribution of 1 symbols is geometric in its y index. [The distribution of $?_{x,y}$ is in fact geometric in y for all $(x,y)\in D$.] This can be justified by noting that two such symbols result from the same set of paths of the process with the corresponding difference only in x or y index of one specific input symbol (for 0's the initial 0 starting the substack and for 1's, the final 1 that falls through to the bottom of the sub-stack). These properties are listed below:

$$\begin{aligned} p(?_{x,y})&=0, \quad x\leqslant -m,\\ p(\diamond_{x,y})&=0, \quad x>0,y>0,\\ p(0_{x,y})&=(1-\rho)^{x-1}p(0_{1,y}), \quad x>0,y\leqslant 0,\\ p(1_{x,y})&=\rho^{y-1}p(1_{x,1}), \quad x\leqslant 0,y>0. \end{aligned} \tag{A5}$$

For a more rigorous proof, it is easy to check that the initial distribution $p_i(\cdot)$ has these properties and that the transition function maintains the properties. Thus, the limiting stationary distribution must also have the properties.

The stationarity recurrence given by Eqs. (A3) and (A4) can be transformed to

$$\frac{\{1-[x>0]p(0)\}p(?_{x,y})}{[1-p(0)](1-\rho)^x\rho^y}=\frac{[(x,y)\in D]p_d(?_{x,y})}{[1-p(0)](1-\rho)^x\rho^y}+\frac{[(x,y)\in E]p_c(?_{x,y})}{[1-p(0)](1-\rho)^x\rho^y}+\frac{p_i(?_{x,y})}{(1-\rho)^x\rho^y}. \tag{A6}$$

For clarity, we define $f_{x,y}$ as the left side of the equation,

$$f_{x,y}:=(1-\rho)^{-x}\rho^{-y}\frac{1-[x>0]p(0)}{1-p(0)}p(?_{x,y}). \tag{A7}$$

This transformation of $p(\cdot)$ cancels out the geometric factors and will decouple the recursive $p(0)$ coefficient from the stationarity equation. The transformation is reversible as $p(0)$ can be obtained in terms of $f_{x,y}$ from

$$p(0)=\sum_{x>0;y}p(0_{x,y})=\sum_{x>0;y}(1-\rho)^x\rho^y f_{x,y}. \tag{A8}$$

With this definition, the geometric properties reduce to

$$\begin{aligned} f_{x,y}&=f_{x,1}, \quad x\leqslant 0,\ y>0,\\ f_{x,y}&=f_{1,y}, \quad x>0,\ y\leqslant 0. \end{aligned} \tag{A9}$$

We define analogously the components $d_{x,y}$ and $c_{x,y}$ corresponding to terms on the right side of Eq. (A6) and apply the definition of $f_{x,y}$ and the above properties,

$$\begin{aligned} d_{x,y}&:=(1-\rho)^{-x}\rho^{-y}\frac{p_d(?_{x,y})}{1-p(0)}\\ &=\sum_{r\geqslant 1}A^r\sum_{a\leqslant 0}\sum_{b\leqslant 0}f_{x-a+rm,b}f_{a,y-b+rk}\\ &=\frac{A}{1-A}\sum_{a\leqslant 0}\sum_{b\leqslant 0}f_{1,b}f_{a,1}, \end{aligned}$$

for $(x,y)\in D$, and

$$c_{x,y} := (1-\rho)^{-x}\rho^{-y}\frac{\rho_c(?_{x,y})}{1-\rho(0)}$$

$$= \sum_{a\leqslant 0}\sum_{b\leqslant 0} f_{x-a,b}f_{a,y-b} = \sum_{a\leqslant 0}\sum_{b\leqslant 0} f_{1,b}f_{a,y-b},$$

for $(x,y)\in E$. The stationarity condition given in Eq. (A6) can then be expanded as

$$\begin{aligned} f_{x,y} &= [(x,y)\in D]d_{x,y} + [(x,y)\in E]c_{x,y} \\ &\quad + (1-\rho)^{-x}\rho^{-y}p_i(?_{x,y}) \\ &= [(x,y)\in D]\frac{A}{1-A}\sum_a\sum_b f_{1,b}f_{a,1} \\ &\quad + [(x,y)\in E]\sum_{a<x;a\leqslant 0}\sum_b f_{1,b}f_{a,y-b} \\ &\quad + [(x,y)\in D]\frac{A}{1-A}, \end{aligned}$$

where we have left out the summation limits for zero terms, based on $f_{a,b}=0$ for $a>0$ and $b>0$.

Noting that only the middle line really depends on x and y and that $f_{a,b}=0$ for $a<-m$, the recurrence can be written as

$$\begin{aligned} f_{x,y} &= [(x,y)\in D]\frac{A}{C} + \sum_{a<x;a\leqslant 0}\sum_b f_{1,b}f_{a,y-b} \\ &\quad - [x>0][y>0]\left(\frac{1-A}{C}-1\right), \end{aligned} \tag{A10}$$

where we define

$$C := \frac{A}{f_{1-m,1}} = \frac{1-A}{\sum_a\sum_b f_{1,b}f_{a,1}+1}, \tag{A11}$$

for reasons to become clear later.

We have thus reduced the stationarity of the stack distribution given by Eqs. (A3) and (A4) to the above convolution equation, where $f_{x,y}$ determines $p(?_{x,y})$ and is given by Eq. (A7).

APPENDIX B: EXACT SOLUTION FOR THE STEADY-STATE FLOW THROUGH GENERATING FUNCTIONS

The convolution recurrence in Eq. (A10), which is the stationarity condition for the stack distributions, can be solved using generating functions (cf. [13]). We define a formal generating function $F_x(z)$ for the sequence f_x, by

$$F_x(z) := \sum_y f_{x,y}z^y. \tag{B1}$$

This generating function is not quite ordinary: the sum goes over all y, positive and negative. In general the values of a generating function do not uniquely determine a sequence that is positive at an infinite distance in both directions. Here, however, we know that both $F_1(z)$ and $F_{1-m}(z)$ are uniquely defined generating functions, because when $x=1-m$, the term $f_{x,y}$ is only positive at an infinite distance in the positive y direction and when $x=1$, it is only positive at an infinite distance in the negative y direction (see Fig. 8). The coefficients for the functions for other x are positive infinitely in both the positive and negative y direction but they are only used in the following calculations for formal multiplication and addition operations corresponding to well-defined convolution and sum operations on sequences.

The term $f_{x,y}$ given by Eq. (A10) vanishes for $x\leqslant -m$. Thus, we can represent it by the first nonzero case $f_{1-m,y}=[y\geqslant 1-k]A/C$, and the differences

$$f_{x,y}-f_{x-1,y} = [x\leqslant 1]\sum_b f_{1,b}f_{x-1,y-b} - [x=1][y>0]\frac{1-C}{C}, \tag{B2}$$

for $x>1-m$. With the generating function notation we have $F_{1-m}(z)=(A/C)z^{1-k}/(1-z)$ and

$$F_x(z)-F_{x-1}(z) = F_1(z)F_{x-1}(z) - \frac{1-C}{C}\frac{[x=1]z}{1-z},$$

for $1-m<x\leqslant 1$. Thus,

$$F_x(z) = [F_1(z)+1]^{x+m-1}F_{1-m}(z) - \frac{1-C}{C}\frac{[x=1]z}{1-z}, \tag{B3}$$

for $-m<x\leqslant 1$. Thus, if we can solve $F_1(z)$ and C, we have determined $F_x(z)$ for all x, because C determines $F_{1-m}(z)$.

For $x=1$, Eq. (B3) can be written as

$$[F_1(z)+1] = [F_1(z)+1]^m\frac{Az^{1-k}}{C(1-z)} - \frac{z-C}{C(1-z)}, \tag{B4}$$

which is an mth degree equation with respect to $\beta := F_1(z)+1$. It is easy to see that there are at most two positive real solutions for β.

Because we can solve $F_1(z)$ given C, the complete solution for the stationary distribution of the stack configurations and thereby ϕ now hinges on determining C. Unfortunately, the quantity C cannot be solved directly from the above equations, since its definition is already used in solving them; equations relating C and $F_1(z)$ reduce to identities when combined with Eq. (B4).

However, there is a different, strange approach: we can determine C from the requirement that $F_1(z)$ must (indirectly) represent a probability distribution. The correct $F_1(z)$ must obviously be analytic for some region $|z|>r$. Additionally, it must be positive and decreasing in z, because it has nonzero coefficients only for nonpositive exponents of z, and all coefficients must be non-negative since they are probabilities multiplied by a positive function of the index. These two constraints allow us to uniquely determine C in the following.

Equation (B4) can be solved with respect to C as

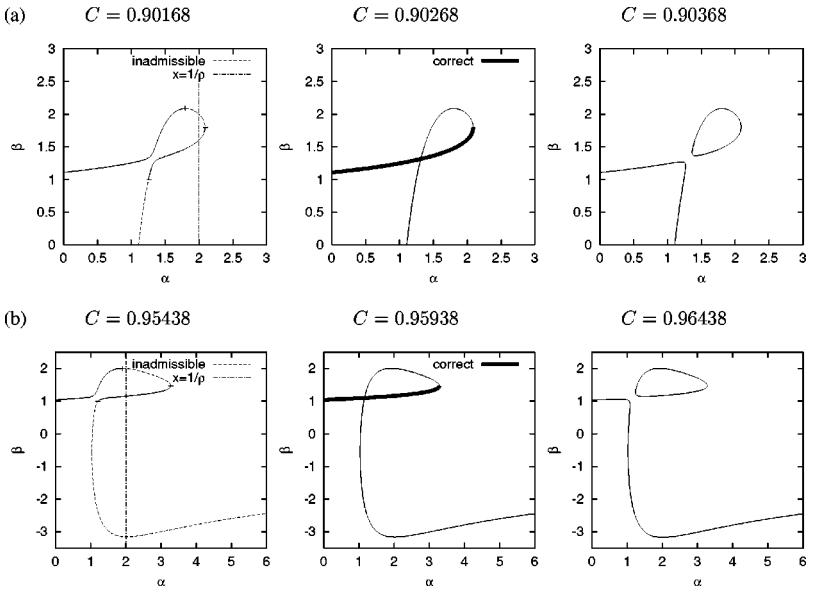

FIG. 9. Plots of the real solutions $\beta = F_1(1/\alpha)+1$ of the generating function equation (B6) at $\rho=1/2$ for C below, at, and above the correct ϕ for (a) $\mathcal{R}_{2,2}$ and (b) $\mathcal{R}_{3,2}$. Because β is an ordinary generating function for a sequence of positive values, the correct solution and all its derivatives must be positive. Furthermore, because $\alpha=1/\rho$ corresponds to a sum of probabilities, β must converge at least for $|\alpha|\leqslant 1/\rho$. This excludes too large values of C. For too small values, the lower curve continues below 1 yielding negative probabilities and the upper curve is not feasible either, because it is decreasing at $\alpha=1/\rho$. Only the singular curve at the correct C yields an admissible solution. Because Eq. (B6) is symmetric with respect to (α,k) and (β,m), the solutions have symmetry axis $\alpha=\beta$ when $m=k$.

$$C=\frac{\beta^m A z^{1-k}-z}{(1-z)\beta-1}, \tag{B5}$$

[note that $\beta=F_1(z)+1$ depends on z]. Substituting $\alpha:=1/z$ in this equation yields a perfectly symmetric form

$$C=\frac{A\alpha^k\beta^m-1}{\alpha\beta-\alpha-\beta}. \tag{B6}$$

Figure 9 depicts the solutions of Eq. (B6) with different values of C for $\mathcal{R}_{2,2}$ and $\mathcal{R}_{3,2}$. The figures are essentially similar for larger m and k with at most two positive solutions and in addition one everywhere negative solution if m is odd. In either case, it can be seen that a too large value of C results to a gap in the solution and a too small value results to either a nonmonotonous or a nonpositive solution. Only the correct C allows changing branches in the singularity point to obtain a feasible solution. This is analogous to the singular behavior of elliptic curves (cf. [16]).

Because the surface $C(\alpha,\beta)$ is smooth, the two constant-C contours can only cross at a critical point. The critical points $\nabla C(\alpha,\beta)=0$ are determined by the equations

$$Ak\alpha^{k-1}\beta^m(\alpha\beta-\alpha-\beta)-(A\alpha^k\beta^m-1)(\beta-1)=0,$$
$$Am\alpha^k\beta^{m-1}(\alpha\beta-\alpha-\beta)-(A\alpha^k\beta^m-1)(\alpha-1)=0.$$

Multiplying by $(\alpha-1)$ and $-(\beta-1)$ and adding, the equations yield $k(\alpha-1)\beta=m(\beta-1)\alpha$. Changing variables to $a=(\alpha-1)/(m\alpha)$ and $b=(\beta-1)/(k\beta)$ yields a simple form $a=b$ for the equation. Substituting $\alpha=1/(1-am)$ and $\beta=1/(1-ak)$ to the critical point equations and to Eq. (B6) results in

$$A(1-am)^{-k}(1-ak)^{-m}=\frac{a}{1-a(k+m-1)},$$
$$A(1-am)^{-k}(1-ak)^{-m}=1-C\frac{1-a(k+m)}{(1-ak)(1-am)}. \tag{B7}$$

Now, we only need to reduce a from this system and then we have an equation relating the unknown C to the parameters A, m, and k. By eliminating the left sides we obtain from the right sides either $a=1/(k+m)$ corresponding to a pole of Eq. (B6) or

$$C[1-a(k+m-1)]=(1-ak)(1-am). \tag{B8}$$

This is a second degree equation for a and its smaller solution is

$$a=\frac{1+(1-C)(k+m-1)-c}{2km}, \tag{B9}$$

where we define

$$c:=\sqrt{[1+(1-C)(k+m-1)]^2-4(1-C)km}. \tag{B10}$$

Note that the other solution with $+c$ in Eq. (B9) does not yield a critical point.

Now Eq. (B8) can be used to rewrite the first equation of Eq. (B7) as

$$A=Ca(1-am)^{k-1}(1-ak)^{m-1} \tag{B11}$$

$$=C^k a[1-a(k+m-1)]^{k-1}(1-ak)^{m-k}, \tag{B12}$$

yielding A as a function of C in closed form with the solution of a given by Eq. (B9).

Now that we have determined all variables, we can determine probabilities for different reactions. The density of new groups on the bottom level of the active stack is

$$\rho_{\Delta g}=\sum_{(x,y)\in D}\sum_{r\geqslant 1} r\sum_{a\leqslant 0}\sum_{b\leqslant 0} p(0_{x-a+rm,b})p(1_{a,y-b+rk})$$

$$=[1-p(0)]\sum_{(x,y)\in D}\sum_{r\geqslant 1} rA^r\sum_{a,b\leqslant 0}(1-p)^x\rho^y f_{1,b}f_{a,1}$$

$$=[1-p(0)]\frac{A}{(1-A)^2}\frac{1-A}{A}\left(\frac{1-A}{C}-1\right).$$

When all levels of the stack are taken into account, the total density of new groups is

$$\rho_{\Delta g}(1+p(0)+p(0)^2+\cdots)=\frac{1}{C}-\frac{1}{1-A}.$$

Adding in the initial density 1 and the density of groups arising from the initial reactions yields the final group density relative to the initial group density:

$$\frac{\rho_G}{\rho(1-\rho)}=1+\frac{A}{1-A}+\left(\frac{1}{C}-\frac{1}{1-A}\right)=\frac{1}{C}.$$

Thus, in the intermediate phase, $\phi=C$.

[1] S. Wolfram, *Cellular Automata and Complexity: Collected Papers* (Addison-Wesley, Reading, MA, 1994).
[2] N. Boccara and H. Fukś, Fundamenta Informaticae (to be published).
[3] N. Boccara and H. Fukś, J. Phys. A **31**, 6007 (1998).
[4] H. Fukś and N. Boccara, Phys. Rev. E **64**, 016117 (2001).
[5] K. Nagel and M. Schreckenberg, J. Phys. I **2**, 2221 (1992).
[6] H. Fukś, Phys. Rev. E **60**, 197 (1999).
[7] H. Fukś and N. Boccara, Int. J. Mod. Phys. C **9**, 1 (1998).
[8] M. Fukui and Y. Ishibashi, J. Phys. Soc. Jpn. **65**, 1868 (1996).
[9] B.-H. Wang, Y.-R. Kwong, and P.-M. Hui, Phys. Rev. E **57**, 2568 (1998).
[10] B.-H. Wang, L. Wang, P. M. Hui, and B. Hu, Phys. Rev. E **58**, 2876 (1998).
[11] Y. Elskens and H. L. Frisch, Phys. Rev. A **31**, 3812 (1985).
[12] K. E. Iverson, *A Programming Language* (Wiley, New York, 1962).
[13] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics*, 2nd ed. (Addison-Wesley, Reading, MA, 1994).
[14] W. Feller, *An Introduction to Probability Theory and its Applications*, 3rd ed. (Wiley, New York, 1968), Vol. 1.
[15] E. Nummelin, *General Irreducible Markov Chains and Nonnegative Operators* (Cambridge University Press, Cambridge, 1984).
[16] A. W. Knapp, *Elliptic Curves, Mathematical Notes* (Princeton University Press, Princeton, NJ, 1992).