Impartial games emulating one-dimensional cellular automata and undecidability

Urban Larsson
Department of Mathematical Sciences
Chalmers University of Technology and Göteborg University
urban.larsson@chalmers.se

Abstract

We study two-player *impartial* games whose outcomes emulate two-state *one-dimensional cellular automata*, such as Wolfram's rules 60 and 110. Given an initial string consisting of a central data pattern and periodic left and right patterns, the rule 110 cellular automaton was recently proved Turing-complete by Matthew Cook. Hence, many questions regarding its behavior are algorithmically undecidable. We show that similar questions are undecidable for our *rule 110 games*. Keywords: cellular automaton, impartial game, rule 110, take-away game, undecidability.

1. Introduction

We study inter-connections between two popular areas of mathematics, *two-player combinatorial games* [BCG04] and *cellular automata* (*CA*) [N66, HU79, W84a, W84b, W84c, W86, W02]. We present an infinite class of games and prove that their *outcomes* (or winning strategies) emulate corresponding one-dimensional CA. In particular we study some recent results of Matthew Cook concerning algorithmic undecidability of Stephen Wolfram's well known *elementary cellular automaton*, rule 110, and interpret these results in the setting of our games. The universality of the rule 110 automaton was conjectured by S. Wolfram in 1985 and proved by M. Cook in [C98, C04, C08]. It is also discussed in the remarkable book [W02].

Our games are played between two players on a finite number of *positions* and are purely *combinatorial*; there is no element of chance and no hidden information. A *rule-set* gives the legal moves of a game. In *normal play*, the ending condition is given by, a player who is not able to move loses and

Preprint submitted to Elsevier *February 26, 2013*

the other player wins. In *misère play*, the winning condition is reversed, a player who is not able to move wins. If the set of options does not depend on whose turn it is, then the game is *impartial*, otherwise the game is *partizan*. It turns out that the game we study can naturally be interpreted either as normal or misère impartial play. Here we have chosen normal play. The outcome classes of these games are denoted P (previous player win) and N (next player win). That is, a position (game) is P if and only if the player whose turn it is to move loses, assuming best play by both players. This gives a recursive characterization of the outcomes of all starting positions of a game and—unless there are drawn positions such as infinite loops where no player can force a win—there will be a partitioning of the set of game positions into N and P.

In this paper, we study two different classes of games with (as we will see) equivalent outcomes. One class is similar to the *take-away* games found in [G66, S70, Z96, L12]. In such games the players take turns in removing tokens (coins, matches, stones) from a finite number of heaps, each with a given finite number of tokens. For certain games the pattern of the two sets of outcomes are reasonably easy to understand. For example it is known that the set of P-positions of a one heap subtraction game, e.g. [BCG04], with a finite number of moves—such as a heap of a finite number of tokens and the rule-set, remove one, two or five tokens—is eventually periodic. On the other hand, in [LW] simple rule-sets are studied that give rise to very complex patterns of P-positions. The classical one heap subtraction games are generalized to several heaps and, by emulating binary one-dimensional cellular automata with finite update functions, it is shown that for finite rule-sets it is undecidable whether or not two games have the same sets of P-positions. It appears that links between CA and two-player combinatorial games are uncommon in the literature. The only sources, except [LW], that we have found so far are [F02, F12]. See also [DH, DH09] for undecidability results of some multiplayer games and for interesting surveys on algorithms, complexity and combinatorial games.

The *cellular automata* use simple rules for updating some discrete structure in discrete (time) steps. For the one-dimensional case we take a doubly infinite binary string as input. As mentioned in the previous paragraph, if we fix some simple initial string (such as a single "1" among "0"s), many problems regarding the behavior of the CA are algorithmically undecidable for finite update functions; e.g. [W02]. In [C04, C08], a particularly simple instance of an update function is studied, Wolfram's rule 110, with the following update function. The value of any given cell remains a "0" if the neighboring cell to the left is also a "0". It remains a "1" if at least one of the two nearest neighbors contains a "0". Otherwise the value switches. If the initial string is arbitrary (or in a sense too simple or too complicated [N66]), for a fixed update function, questions of decidability do not appear interesting. It turns out that a natural setting for the rule 110 CA is to code the program in a central finite part of the initial string together with certain periodic left and right patterns. Under these assumptions it is shown, in [C04, C08], that many questions regarding the rule 110 CA are undecidable. In fact, universal Turing machines with the least known number of states and symbols were constructed by simulating the rule 110 automaton. In contrast the simpler Wolfram's rules 60, defined in Section 2, and 90 are known to be decidable under the above assumptions.

Since the one-dimensional cellular automata generate two-dimensional patterns over 'time', a reasonable goal would be to try and emulate the CA, by games, in no more than two dimensions. We will see that this can be done, in the *triangle-placing game*, by two players alternately placing *isosceles right triangles* on the upper half plane according to certain rules. In the setting of take-away games we will see that indeed our games will be played on two finite heaps, a *time-heap* and a *tape-heap*, a terminology introduced in [LW] (but where several heaps were used in the simulation of cellular automata). Our variation of take-away games belongs to a different family than the classical subtraction games, namely the number of tokens a player is allowed to remove depends in some way on the previous player's move; so that in fact our game positions will be represented by ordered triples of non-negative integers. Such games are sometimes called *move-size dynamic* e.g. [L09] and in fact, we will adapt an idea from that paper: the move options on one of the heaps will depend on the other player's move on the other heap (although the context is quite different here).

In Section 2, as an introductory example, we begin by studying a take-away game that emulates Wolfram's rule 60 CA and in a particularly simple setting, namely where the outcomes form Pascal's triangle modulo 2. In Section 3, we define triangle-placing games that emulate an infinite class of CA. Then, in Section 4, we generalize the setting in Section 2 and define a class of take-away games with equivalent outcomes as the triangle-placing games. In Section 5, we discuss how some undecidability problems for the rule 110 cellular automaton are interpreted in our setting of combinatorial games.

2. The rule 60 game and Pascal's triangle

A position (X, Y, m_p) of the *rule 60 game* consists of a finite heap of $X \geq 0$ tokens and a finite heap of $Y \geq 0$ matches. The matches simulate 'time'. There are 2 players who alternate removal of tokens and matches according to the following rules. A move consists of two parts: (1) at least one match is removed, at most the whole heap; (2) at most m_p tokens are removed (possibly none), where $m_p > 0$ denotes the number of matches removed by the other player in the previous move. A player may not remove the whole heap of matches, unless all tokens are also removed. A player who is unable to move loses. The other player wins. See Figures 1 and 2.

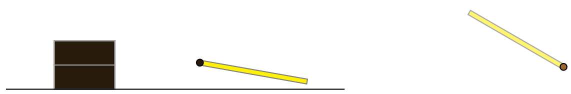

Figure 1: The previous player removed the rightmost match. Hence at most one token may be removed, which means that no move is possible and hence the previous player wins.

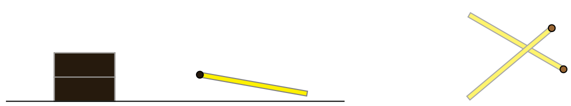

Figure 2: In this game, the next player wins by removing the last match together with both tokens.

The update rule of Wolfram's rule 60 cellular automaton is as follows. Assign arbitrary binary digits to a_x^0 for all integers x. For $y > 0$, let $a_x^y = 0$ if $a_{x-1}^{y-1} = a_x^{y-1}$ and otherwise let $a_x^y = 1$. In other words $a_x^y = f(a_{x-1}^{y-1}, a_x^{y-1})$, where $f(i, j) = i \oplus j$, the operation being binary addition without carry, the *Xor gate*. The two-dimensional patterns obtained by this cellular automaton are algorithmically decidable given that the initial one-dimensional pattern is sufficiently simple, say left and right periodic together with a central data program. In particular, if a spatial pattern consists in a single "1", say $a_i^1 = 1$ if and only if $i = 1$, then the updates correspond precisely to Pascal's triangle modulo 2. In fact, the outcomes of the rule 60 game correspond to the updates of the rule 60 CA with an initial string of the form $\ldots 000111 \ldots$, see Figure 3. Our first result is a special case of Theorem 4.1 in Section 4.

Theorem 2.1. *Let the initial condition of the rule 60 CA be $a_i^0 = 1$ if and only if $i > 0$. Then, a position (X, Y, m_p), in the rule 60 game, with $X \geq 0$ tokens, $Y \geq 0$ matches and where the previous player removed $m_p > 0$ matches is a second player win if and only if $a_X^Y = \cdots = a_X^{Y+m_p-1} = 0$ and if $Y > 0$ then $a_X^{Y-1} = 1$.*

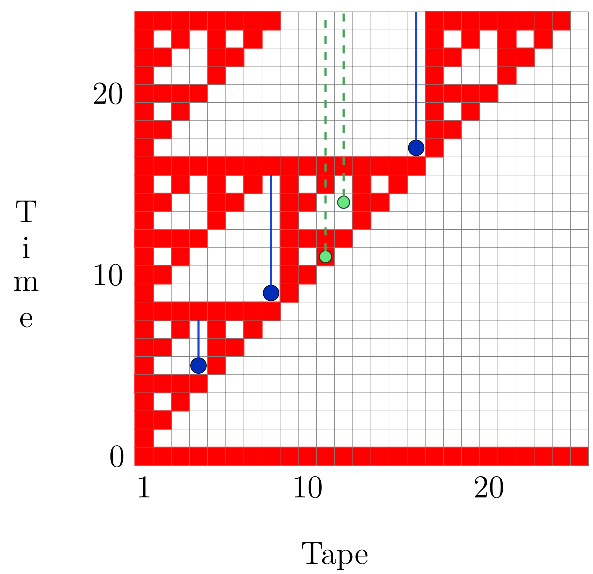

Figure 3: The CA given by $f(x, y) = x \oplus y$ (Wolfram's rule 60) together with an initial string of the form $\ldots 0011\ldots$, the "1"s correspond to dark (red) cells and the initial "0"s are omitted. (Note that time flows upwards.) By Theorem 2.1, for example the dark (blue) "circle+tail"s, corresponding to positions $(4, 5, 3), (8, 9, 7), (16, 17, 15), \ldots$, are all P (the latter extends above the figure), whereas the small light (green) positions $(11, 11, m_p)$ and $(12, 14, m_p)$ are N, for all m_p. See also Table 1.

X	2	2	2	3	3	3	3	4	4	4	4	4	4	5	5	5	5
Y	1	3	5	1	1	5	5	1	1	1	5	5	5	1	1	1	1
m_p	1	1	1	1	2	1	2	1	2	3	1	2	3	1	2	3	4

Table 1: A list of all P-positions (X, Y, m_p), for $X \leq 5$ and $Y \leq 5$, of the rule 60 game, except those of the form $(0, Y, m_p)$, which are P if and only if $Y = 0$. Note that a triple of the form $(X, 0, m_p)$ is a legal game position if and only if $X = 0$.

3. An impartial triangle-placing game

We will connect the outcomes of our games to the patterns of cellular automata. Let us begin by defining the CA of interest; see Figure 4 for some examples and also Figure 5. Assign an initial string of binary values $A = (a_x) = (a_x^0)$ for all integers x. The update rule of the cellular automaton CA(A, Γ, γ), for γ and Γ non-negative integers, is as follows: For $y > 0$, let $a_x^y = 0$ if

$$a_{x-1}^{y-1} = a_x^{y-1} = 0$$

or if

$$a_{x-\Gamma-1}^{y-1} = \cdots = a_{x+\gamma}^{y-1} = 1$$

and $a_x^y = 1$ otherwise. (Then $\Gamma = \gamma = 0$ correspond to rule 60.)

Our game is played on the *upper half plane*, which consists of all ordered pairs of integers (x, y) with $y \geq 0$. When we use the term *triangle* in this section we think of the set of discrete lattice points that are covered by a certain triangle shape with horizontal base and in case of an *isosceles right triangle*, or *IRT*, the right angle is to the right. In particular, a *triangle position* (x, y, h), h a positive integer, is an IRT in the upper half plane, which covers the point $(x, y + h - 1)$ at the top and the set $\{(x - h + 1, y), \ldots, (x, y)\}$ at the base; Figure 6. Thus, if $h = 1$ we have a trivial IRT, covering a single point. Yet, it will be convenient to think of h as the height of the triangle position. Also its *support* has size $\Gamma + h + \gamma + 1$ and covers $\{(x - \Gamma - h, y - 1), \ldots, (x + \gamma, y - 1)\}$. Here we only require that the triangle position is contained in the upper half plane, that is that $y \geq 0$, so that it is legal for a (final) triangle position to have its support at the y-coordinate -1.

The rules of the *triangle-placing game* T(A, Γ, γ) are as follows. Let (A, Γ, γ) be as in CA(A, Γ, γ). Two players alternate in placing IRTs on the upper half plane. Suppose that the triangle position (x, y, h) is given. Then the next player places another IRT, say (x', y', h'), of the same shape but possibly different size, with its top intersecting the support of the previous triangle, that is satisfying $x' \in \{x - (\Gamma + h), \ldots, x + \gamma\}$, with $y' + h' = y$ and $1 \leq h' \leq y$; Figure 8. Note that by the rules of game, the actual 'game board' is bounded by a shape as in Figure 6. (See also the first part of the proof of Theorem 4.1.)

The *ending condition* is provided by the doubly infinite binary string $A = (a_x)$. A player can place the triangle $(x, 0, h)$ if and only if $a_{x-h+1} +$

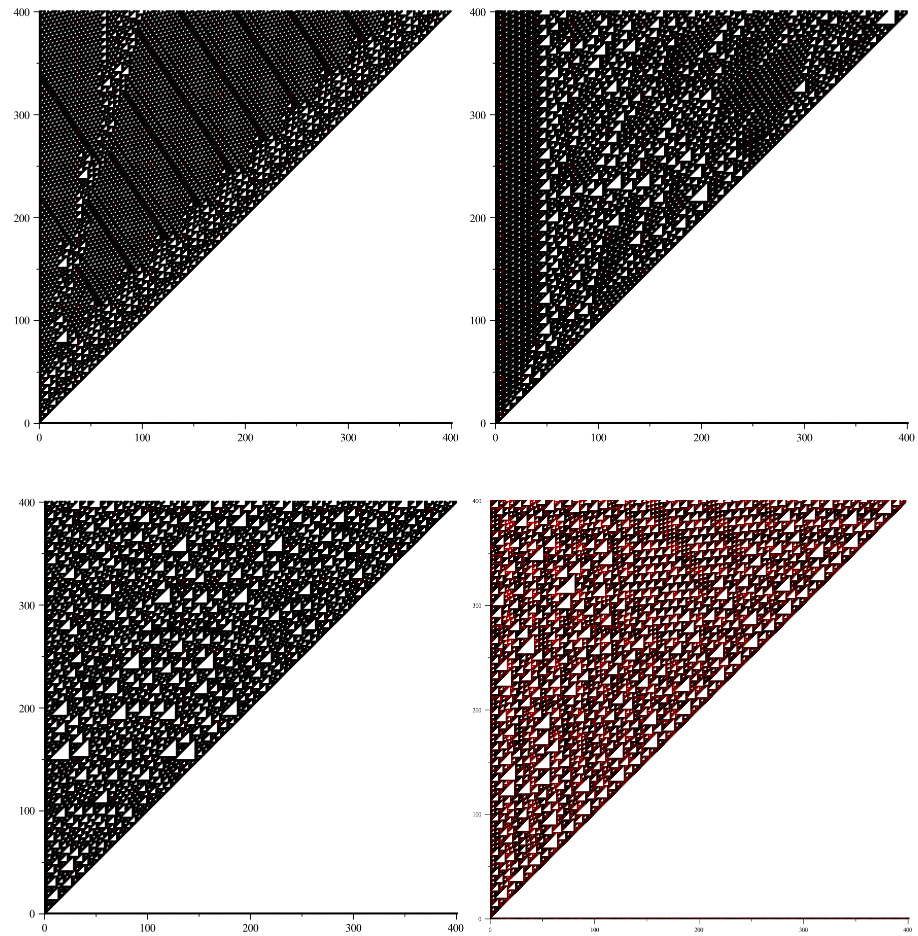

Figure 4: The updates of CA($A, 0, 1$), CA($A, 2, 1$), CA($A, 1, 1$) and CA($A, 0, 3$), for $A = \ldots 0011 \ldots$ and $x, y \leq 400$. They all appear to have non-trivial behavior. A more definite common feature is the reappearance of patterns of white cells ("0"s) in the shape of isosceles right triangles. As for rule 60 in Figure 3 this is just a simple consequence of their respective update functions.

$\cdots + a_x = 0$. Hence, the ending condition does not depend on Γ and γ. By the update rule of the CA, another way of stating the ending condition is to require that if $y = 0$ then the final IRT covers only "0"s in the underlying CA-cells. Then, the game ends at the level $y > 0$, if and only if $y = 1$ and there is no legal move to level $y = 0$; Figures 9 and 10.

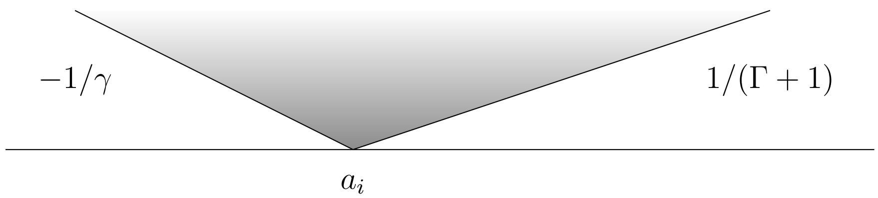

Figure 5: The time-wise influence of the initial CA value a_i is bounded below by lines of slopes $-1/\gamma$ and $1/(\Gamma+1)$ respectively.

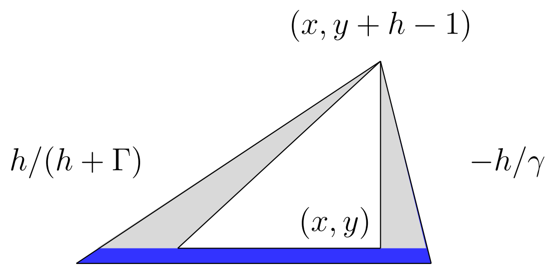

Figure 6: The triangle position (x, y, h) is covered by an isosceles right triangle (in white) with (x, y) at the right angle and $(x, y+h-1)$ at the top. Its support in dark blue is $\{(x-h-\Gamma, y-1), \ldots, (x+\gamma, y-1)\}$. The slopes of the sides connecting the support to the top are $h/(h+\Gamma)$ and $-h/\gamma$ respectively.

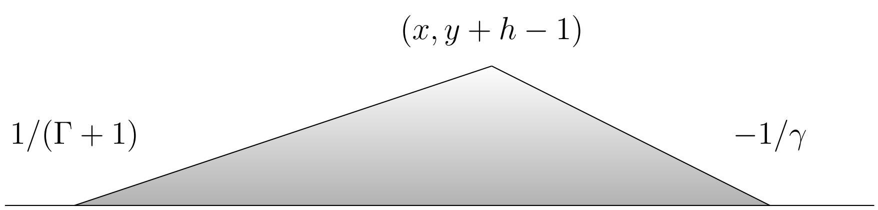

Figure 7: The outcome of a triangle position (x, y, h) is influenced by the CA-cells bounded above by lines of slopes $1/(\Gamma+1)$ and $-1/\gamma$ respectively.

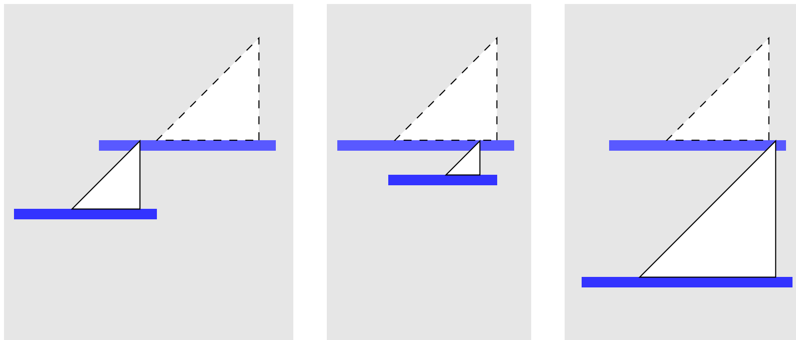

Figure 8: Three typical moves in the triangle placing game. The dashed IRTs indicate the previous triangle positions for the respective games. In each case, by the rules of game, the top of the current triangle position intersects the support of the previous one.

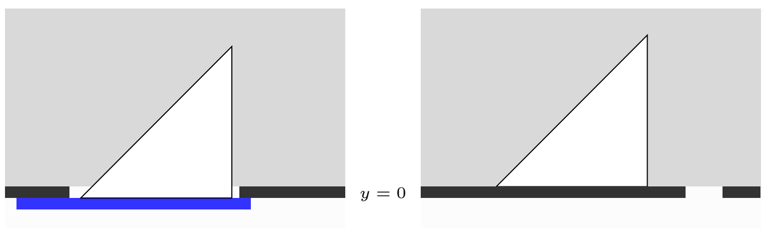

Figure 9: Two final triangle positions; to the left, the triangle position's support does not belong to the upper half plane, so no move is possible; to the right, there are no white CA-cells, at the terminal level $y = 0$, within the support of the triangle position.

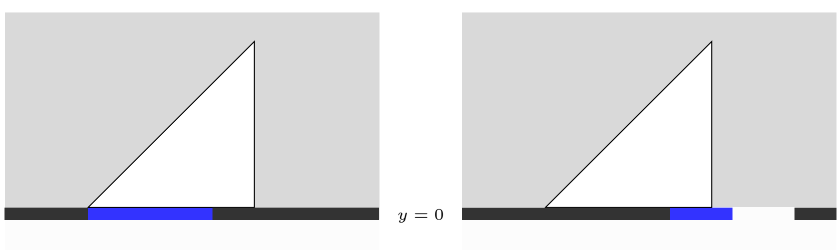

Figure 10: These triangle positions are non-terminal and losing since the next player can place a final IRT of height 1, within the support of the current triangle position, at the terminal level $y = 0$.

Theorem 3.1. *Given $T(A,\Gamma,\gamma)$, the triangle position (x,y,h) is P if and only if it covers only "0"s and, if $y>0$, its* support *covers only "1"s in the update of $CA(A,\Gamma,\gamma)$, or equivalently $a^y_{x-h+1}+\cdots+a^y_x=0$ and $a^{y-1}_{x-(\Gamma+h)}+\cdots+a^{y-1}_{x+\gamma}=\Gamma+h+\gamma+1$.*

Proof. We need to show that, if a triangle position covers only "0"s, then none of its options does (denoted by "P→ N") and that if a triangle position covers at least one "1", at least one of its options covers only "0"s (denoted by "N→ P").

For the "$P\to N$" direction, if $y=0$ we are done so suppose that $y>0$. If the triangle (x,y,h) covers only "0"s and its support covers only "1"s, then by the update of the CA, the next IRT covers one of these "1"s.

For the "$N\to P$" direction, suppose that the triangle position (x,y,h) covers a "1", say in cell (r,s). Then at least one of the two CA-cells $(r,s-1)$ or $(r-1,s-1)$ must contain a "1". Hence, we can iterate this process of detecting "1"s covered by the triangle position until we approach its base. Since this base covers a "1", by the ending condition this gives $y>0$, and hence, by the update rules of the CA, the support of the triangle position (x,y,h) must cover a "0". By the rules of triangle-placing, the next player can use the "0"-cell, at y-coordinate $y-1$, as the top of the next IRT and, by the update rules of the CA, choose it carefully as to only cover "0"s and simultaneously guarantee only "1"s underneath its support if its y-coordinate is positive. □

In this proof the intention has been to convey the main idea of how our games emulate the desired cellular automata. For an analogous proof, covering some more detail, see the second part of the proof of Theorem 4.1.

4. Take-away games and cellular automata

Let us next define a generalization of the *take-away game* of rule 60 from Section 2, where, as in the previous section, the move options depend on two non-negative integer parameters, Γ and γ, and where the ending is conditioned on a *black* or *white coloring* of each token. Let $\tau=\tau_1\ldots\tau_X$ denote a finite binary string. Then the ith token is black if and only if $\tau_i=1$, the first token is at the bottom of the heap and the Xth token is at the top. It may be convenient to think of 'non-positive tokens' as white, but we will make our definitions independent of this.

Hence a game position $(X, Y, m_p) = (\tau, X, Y, m_p)$ consists of a finite *tape-heap* of X ordered and τ-colored tokens and a finite *time-heap* of Y (unordered) matches, where X and Y are non-negative integers and where $m_p > 0$ denotes the number of matches removed by the other player in the previous move. Since our take-away games are move-size dynamic, the move options for the next player will depend in some precise manner on the previous player's move.

Suppose that a position (τ, X, Y, m_p) is given together with the game parameters Γ and γ. The two players alternate turns in which they, at each stage of the game, remove $0 \leq t \leq X$ tokens from the top of the tape-heap and $1 \leq m \leq Y$ matches from the time-heap according to the following rules.

(I) The number t of tokens removed from the tape-heap must satisfy $0 \leq \gamma(m-1) \leq t \leq \gamma m + m_p + \Gamma$, with the exception that if the number of remaining tokens is less than $\gamma(m-1)$ then all of them can be removed.

(II) By $m \geq 1$, at least one match has to be removed. The whole heap of Y matches can be removed if and only if there is no black token among the top $\min\{X, Y\}$ tokens.

Hence, if there are no matches left, a player cannot move and the other player wins. But the game can also end because it is not possible to remove a final single match as described in (II).

Since we are interested in emulating one-dimensional cellular automata and the triangle-placing game in the previous section, we wish to have a mechanism to simulate a doubly infinite binary string from our finite tape-heap. For this purpose we make the following somewhat technical definition. (The move rules in any specific game are given in (I) and (II) and do not depend on this paragraph.) Given game constants Γ, γ and a doubly infinite binary string A, we denote a game family by $\mathrm{G}(A, \Gamma, \gamma)$ and a specific game by $\mathrm{G}(A, \Gamma, \gamma)_{\xi}^{(X,Y,m_p)}$, where ξ together with the string A, determine the specific coloring of the game. Precisely, the X tokens are colored by the finite binary string $a_{\xi+1} \ldots a_{\xi+X} \subset A$ via the rule: $a_{\xi+i} = 1$ iff the ith token is black, that is $\tau_i = a_{\xi+i}$, for all i. The bottom token is colored according to the value of $a_{\xi+1} = \tau_1$ and the top token according to $a_{\xi+X} = \tau_X$.

If $A = \underline{0}$ then the first player wins (independently of the other variables), where the underscore denotes a periodic given pattern (infinite or doubly infinite). As we have seen in Section 2, if $A = \underline{0}\,\underline{1}$, precisely $a_x = 1$ if and only if $x \geq 1$, $\xi = 0$ and $\Gamma = \gamma = 0$, the CA describes the outcomes of the rule

60 game. In the next section we study the *rule 110 game* which corresponds to $\Gamma = 0$ and $\gamma = 1$. See also Figure 11 for an illustrative example of this game. Its outcome is illustrated in Figure 12 and Table 2, which leads us to the main result of this section.

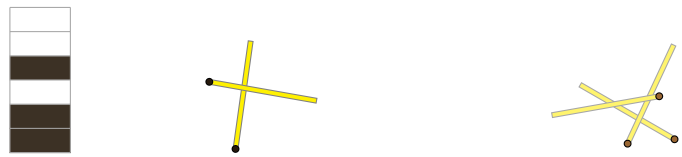

Figure 11: Here we exemplify the rule 110 game via the position $(110100, 6, 2, 3)$. Note that, since $\gamma = 1$, the first player cannot win by removing both matches. Namely, they have to be accompanied by at least one and at most five tokens. But, by the coloring of the tape-heap, such moves are not legal. Another possibility for a next player win would be to remove one match and $0 \le t \le m_p = 3$ tokens. But, by the coloring of the tape-heap, this gives the second player the opportunity to remove the single match and either zero or one token and win. Hence, the position is a second player win, which also follows by Theorem 4.1 and Figure 12.

Given a take-away game, $\mathrm{G}(A, \Gamma, \gamma)_0^{(X,Y,m_p)}$, it will be convenient to think of the number of tokens as a linear translation from the (x, y)-coordinates of the corresponding $\mathrm{CA}(A, \Gamma, \gamma)$, $X = \varphi(x, y) = x + \gamma y$.

Theorem 4.1. *Let $A = (a_i)$ denote an initial condition of the cellular automaton $CA(A, \Gamma, \gamma)$ and let the game be $G(A, \Gamma, \gamma)_0^{(X,Y,m_p)}$. If, in addition, $X \ge (\Gamma + \gamma + 1)Y + m_p$, then the following conditions are equivalent.*

(i) The updates of the CA satisfy $a_x^y = \cdots = a_x^{y+m_p-1} = 0$ and if $y > 0$ then $a_x^{y-1} = 1$.

(ii) The game position $(X, Y, m_p) = (\varphi(x, y), y, m_p)$ is P, a previous player win.

The same result holds in full generality with the initial condition of the CA exchanged for $A = \ldots 00a_1a_2\ldots$. In particular this result holds whenever $-\gamma y \le x < (\Gamma + 1)y + m_p$.

Proof. We begin by proving that if

$$X \ge (\Gamma + \gamma + 1)Y + m_p, \tag{4.1}$$

then the game will end with sufficient number of tokens in the tape-heap, so that the coloring of the tokens determines the ending condition. In this way we can guarantee that the outcomes are independent of the non-positive part of the CA's initial condition; see also Figures 5 and 7. Suppose that the position is (X, Y, m_p) and that $(t_i)_{i=1}^n$ and $(m_i)_{i=1}^n$ represent the complete sequences of removal of tokens and matches respectively until the end of the game, with $n \leq Y$ the number of entries in the sequences and where n indicates the last move. Then, by definition, the total number of removed tokens satisfies

$$\sum_{i=1}^{n} t_i \leq \gamma \sum_{i=1}^{n} m_i + m_p + \sum_{i=1}^{n-1} m_i + n\Gamma, \tag{4.2}$$

where $\sum_{i=1}^n m_i \in \{Y-1, Y\}$, by the ending condition of the game. It follows that the right hand side of the inequality is maximized for $m_n = 1$ and $n = Y$. For this "worst" case, in each move, except the first one, $\gamma + \Gamma + 1$ tokens are removed. In the first move, $\gamma + \Gamma + m_p$ tokens are removed. Hence, if $X > \gamma Y + m_p + Y - 1 + Y\Gamma$, then the coloring of the tape-heap will determine the outcome of the game, which proves the claim.

For the rest of the proof we can ignore the finiteness of the heap of tokens and assume the appropriate initial condition of the CA. Hence, as indicated above, we can identify the outcome of our heap game with that of the triangle-placing game, by setting $m_p = h, (X, Y) = (\varphi(x, y), y)$. Namely. the result follows since (i) is equivalent to the triangle position covering only "0"s and its support covering only "1"s. However, we have promised to give a somewhat more detailed variant of the proof in this setting.

In analogy with the setting of the triangle-placing game, we need to show that, if a position as in (ii) satisfies (i), then none of its options does (denoted by "P→ N") and that if a position as in (ii) does not satisfy (i) then one of its options does ("N→ P"). Let us begin with the former case.

"P→N": Suppose first that $(\varphi(x, y), y, m_p)$ is of the form in (i). Then we need to show that none of its options is of this form. We may assume that $y > 0$ since otherwise there is no option. Let $1 \leq m \leq y$. An option is of the form

$$(\varphi(x', y'), y', m) = (\varphi(x, y) - t, y - m, m) \tag{4.3}$$

where $0 \le \gamma(m-1) \le t \le \gamma m + m_p + \Gamma$, which gives $t = x - x' + \gamma(y - y') = x - x' + \gamma m$ and so $\gamma(m-1) \le x - x' + \gamma m \le \gamma m + m_p + \Gamma$ which implies

$$x - m_p - \Gamma \le x' \le \gamma + x. \tag{4.4}$$

By the assumption we have that $a_x^{y-1} = 1$. Also, item (i) together with the updates of the CA give $a_{x-i}^y = 0$ for $0 \le i \le m_p - 1$. Altogether this gives that

$$a_{x'}^{y'+m-1} = 1 \tag{4.5}$$

for all $x - m_p - \Gamma \le x' \le \gamma + x$, which by (4.4) proves the claim.

"$N \to P$": For this case we have to show that it is possible to find an option of the form in (i) whenever one is playing from a position not of this form. Suppose first that $a_x^{y-1} = 0$. Then there is a least $i \ge 1$ such that $a_x^{y-i} = 0$ and ($a_x^{y-i-1} = 1$ or $y = i$). Remove $i = m$ matches and γm tokens, all tokens if $x < \gamma m$. Then the new position is of the correct form (since we assume that $a_{x-\gamma m} = 0$ if $x < \gamma m$).

Otherwise we may assume that $a_x^{y+i} = 1$ for some least $0 \le i \le m_p - 1$. By the updates of the CA (and minimality) this gives that the cell

$$a_{x-i}^y = 1, \tag{4.6}$$

but $a_{x-i+j}^y = 0$, for all $1 \le j \le i$. We may assume that $a_x^{y-1} = 1$, which then, by the update rules of the CA, implies $a_{x-j}^{y-1} = 1$, for all $0 \le j \le i + \Gamma$. Therefore, by (4.6), the update rules of the CA force $a_{x-i-\Gamma-1}^{y-1} = 0$. Hence it suffices to remove tokens so that $x' = x - i - \Gamma - 1$ is the x-coordinate of the CA-cell corresponding to the new position; that is $\gamma m + i + \Gamma + 1$ tokens, where m is the number of removed matches. By the game rules (I), we need to check that $-\gamma \le i + \Gamma + 1 \le m_p + \Gamma$. By definition of i, the lower bound is clear since $i \ge 0$ and the upper bound, since $i + 1 \le m_p$. □

In Theorem 4.1, for convenience, we have implicitly set $\xi = 0$, but one can easily deduce that it holds for all ξ since the string A can be translated arbitrarily preserving the same time-wise CA patterns (but at different spatial locations). The following corollary of Theorem 4.1 supplies information about this and about the N-positions of our games.

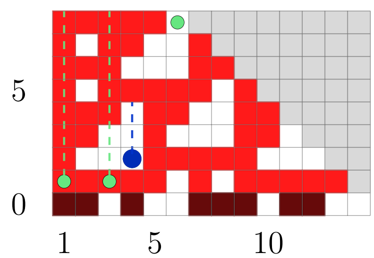

Figure 12: The pattern represents a few initial CA updates for rule 110 for the initial bit string, $\tau =$"11010011101100" (and an infinite string of 0s to the left). The updates also indicates winning strategies for the corresponding rule 110 game's heap positions (m_p dashed) for the ending condition τ. For example, the large dark (blue) circle represents $(110100, 6, 2, m_p)$. It is a second player win if and only if $1 \le m_p \le 3$. For the smaller light (green) circles the first player wins independent of m_p. For example, for the leftmost green position it is always possible to remove precisely the final match and the final token. The cells in the gray area do not affect the outcomes of the given positions.

x	2	2	2	3	3	3	4	4	4	4	5	5	5	6	6	6	6	6	6
y	2	5	7	0	2	2	2	2	2	6	0	6	6	0	0	3	6	6	6
X	4	7	9	3	5	5	6	6	6	10	5	11	11	6	6	9	12	12	12
Y	2	5	7	0	2	2	2	2	2	6	0	6	6	0	0	3	6	6	6
h, m_p	1	1	1	1	1	2	1	2	3	1	1	1	2	1	2	1	1	2	3

Table 2: This is a list of some P-positions of our rule 110 games (that is $\Gamma = 0$ and $\gamma = 1$). The triangle-placing game's P-positions are denoted (x, y, h), whereas the take-away game's P-positions are (X, Y, m_p). The initial string of the rule 110 CA is "11010011101100" as in Figure 12. We show all P-positions for $0 < x \le 6$ and $0 \le y \le 6$. (Positions of the form $(0, Y, m_p)$ are P if and only if $Y = 0$.) See also Corollary 4.3.

Corollary 4.2. *A position $(\varphi(x, y), y, m_p)$, of the game in Theorem 4.1, is N if and only if the corresponding CA updates satisfy one of the following:*

(a) $a_x^{y+i} = 1$ for some $i \in \{0, \dots, m_p - 1\}$ or

(b) $a_x^{y-1} = 0$.

Shift the indices in A by ξ steps so that, in particular the content of the new 0-cell becomes that of the old ξ-cell, that is define $a'_{x-\xi} = a_x$ for all x. Then Theorem 4.1 and the first paragraph of this corollary hold with A exchanged for (a'_i) and each x exchanged for $x - \xi$.

If the condition $X \geq (\Gamma+\gamma+1)Y+m_p$, in Theorem 4.1, is satisfied, where the number of tokens in the tape-heap is X and the number of matches in the time-heap is Y, then we say that the tape-heap is *super-critical*. See also Figure 6. The point of making this definition is that many results translate immediately between the CA and the take-away game for super-critical tape-heaps.

Let us formally state the correspondence of the outcomes of our two families of games.

Corollary 4.3. *Let the game parameters Γ and γ be given. Suppose further that x, y, h, X, Y, m_p are integers with $Y = y \geq 0$, $m_p = h > 0$ and where the tape-heap with $\varphi(x, y) = X \geq 0$ tokens is super-critical. Then, the position (x, y, h) in the triangle-placing game is P if and only if (X, Y, m_p) is P in the take-away game.*

Two consequences of the above results are the following 'periodicity lemma' and the subsequent 'convergence lemma'.

Lemma 4.4. *Let $A = (a_i)$ denote a doubly infinite binary string. Then the following conditions are equivalent.*

- *The CA(A, Γ, γ) has* two-dimensional eventually periodic updates*, that is, there is a finite number of classes of intersections with rational polyhedra (one of them bounded) such that, for each class there is a universal pair of constants (δ, ρ), such that, for all x and y in this class, $a_x^y = a_{x+\delta}^{y+\rho}$.*

- *The games in $T(A, \Gamma, \gamma)$ and the super-critical games in $G(A, \Gamma, \gamma)$ have* two-dimensional eventually periodic outcomes*: that is, there is a finite number of classes of intersections with rational polyhedra, with associated pairs of universal constants (ρ', δ') such that, for each class, for all ξ, for all X, Y and m_p, the outcomes of the positions (X, Y, m_p) and $(X + \delta', Y + \rho', m_p)$ are identical.*

Proof. By Corollary 4.3 it suffices to give a proof for, say the take-away game. For simplicity, by Corollary 4.2, we may assume $\xi = 0$. Suppose that the CA has two-dimensional periodic patterns in some class, spatially and timewise with period δ and ρ respectively. By Theorem 4.1 we get that the outcomes of the positions $(\varphi(x, y), y, m_p)$ and $(\varphi(x + \delta, y + \rho), y + \rho, m_p)$ are identical. Hence we can take $\delta' = \varphi(\delta, \rho)$ and $\rho' = \rho$.

Suppose, on the other hand, that the outcomes of the positions $(X, Y, m_p) = (\varphi(x, y), y, m_p)$ and $(X+\delta', Y+\rho', m_p) = (\varphi(x, y)+\delta', y+\rho', m_p)$ are identical with super-critical tape-heaps. If they are both P, then, by Theorem 4.1, we have that $a_x^y = a_{x+\delta'-\gamma\rho'}^{y+\rho'} = 0$. Further, by definition of P, for all $0 \le i < m_p$, we also get $a_x^{y+i} = a_{x+\delta'-\gamma\rho'}^{y+i+\rho'} = 0$. This implies that if a pattern of N-position is periodic and at least one of them is of type (b) in Corollary 4.2, then all N-positions in this pattern are of type (b). Otherwise we get that $a_x^y = a_{x+\delta'-\gamma\rho'}^{y+\rho'} = 1$. In either case we can take $\delta = \delta' - \gamma\rho'$ and $\rho = \rho'$. Since we have assumed super-critical tape-heaps, the values of the CA correspond precisely to those of the games according to Theorem 4.1. □

The method in the proof actually says that the (three-dimensional) game positions define the pattern of the corresponding CA uniquely via its set of P-positions. This observation is used again in the next result. Let $A = (a_i)$ and $B = (b_i)$ denote doubly infinite binary strings. We say that the games in $\mathrm{T}(A, \Gamma, \gamma)$ and $\mathrm{T}(B, \Gamma, \gamma)$ *converge* if, for all sufficiently large (x, y), their respective outcomes of the triangle position (x, y, h) are the same, for all h. We say that the games in $\mathrm{G}(A, \Gamma, \gamma)$ and $\mathrm{G}(B, \Gamma, \gamma)$ *converge* if, for all games on sufficiently large time-heaps with super-critical tape-heaps, for all ξ and m_p, their outcomes are identical. The cellular automata $\mathrm{CA}(A, \Gamma, \gamma)$ and $\mathrm{CA}(B, \Gamma, \gamma)$ *converge* if and only if, for all sufficiently large y, $a_x^y = b_x^y$ for all x.

Lemma 4.5. *Let $A = (a_i)$ and $B = (b_i)$ denote doubly infinite binary strings. The games in $T(A, \Gamma, \gamma)$ and $T(B, \Gamma, \gamma)$ converge if and only if the games in $G(A, \Gamma, \gamma)$ and $G(B, \Gamma, \gamma)$ converge if and only if the cellular automata $CA(A, \Gamma, \gamma)$ and $CA(B, \Gamma, \gamma)$ converge.*

Proof. The correspondence of the games is clear by Corollary 4.3. Since the tape-heaps are super-critical, by Theorem 4.1, the outcomes for the respective games in a family correspond precisely to the patterns of the corresponding CA. Hence, by Theorem 4.1, if the CA converge, it follows that the outcomes of the game families converge. For the other direction we use a similar argument as in Lemma 4.4, the patterns of the CA is defined uniquely, given only the description of the P-positions of the game, via the move-size dynamic rule. □

5. The rule 110 game and undecidability

In this section, we look into questions of decidability for the outcomes of our games. The results may equivalently be interpreted in the setting of the triangle placing games or the take-away games. By Lemma 4.4, it is decidable whether the updates of any of our CA eventually become two-dimensional periodic if and only if it is decidable whether the outcomes of the corresponding game do. By Lemma 4.5, it is decidable whether two games converge if and only if it is decidable whether the corresponding CA converge. As we discussed briefly in the introduction, questions of algorithmic decidability requires a finite input. For the CA, this is achieved by letting the initial binary string be doubly periodic with a finite central data pattern. Such a binary string can encode the ending condition of a (family of) game(s), as described in previous sections.

By recent results of Matthew Cook [C04, C08] Wolfram's rule 110 CA, which in our notation is $\mathrm{CA}(A, 0, 1)$, is particularly interesting, and hence also the games $\mathrm{T}(A, 0, 1)$ and $\mathrm{G}(A, 0, 1)$, called the *rule 110 games*. Let the initial binary string of this CA be of the form $A = \underline{L}C\underline{R}$, where L, C and R are finite binary strings (or equivalently integers coded in binary) and where, as before, underscore denotes a periodic pattern. Cook proved the following results.

Theorem 5.1 ([C04, C08]). *For finite binary strings L and R and a central finite data string C, it is algorithmically undecidable whether the rule 110 CA with $\underline{L}C\underline{R}$ as input ever produces a given binary string.*

The proof uses that the time-wise binary string "110101010111111" (or equivalently the spatial binary string "01101001101000") is produced if and only if a certain *F-glider* is created in the interaction of other *gliders* from the updates of the periodic L-pattern and the central C-pattern; Figures 13 and 14. Using *cyclic tag-systems* [C04, C08] a universal Turing machine is programmed to halt if and only if the given binary string occurs in the updates of the CA. This is how the rule 110 CA is proved undecidable. One consequence of this result is that it is undecidable if the patterns in this CA will ultimately become two-dimensional periodic. If the central program goes into a loop, then the updates of the CA will consist of a finite number of 2-dimensional periodic regions connected with (one dimensional) periodic 'seams' that will never meet. By Lemma 4.4, an analogous corollary holds for our games.

Corollary 5.2 ([C04, C08])**.** *Let L, R and C denote finite binary strings. It is algorithmically undecidable whether (the central data pattern of) the rule 110 CA with $\underline{L}C\underline{R}$ as input becomes two-dimensional eventually periodic.*

Corollary 5.3. *For fixed binary strings L, R and a central data pattern C, it is algorithmically undecidable whether the outcomes of the rule 110 games, with ending condition given by $\underline{L}C\underline{R}$, are two-dimensional eventually periodic.*

It is easiest to consider the triangle-placing game here, but Corollary 4.3 obviously applies so that the same result holds for the take-away game.

The halting problem for a universal Turing machine can also be translated to the setting of our games via a finite path of alternating moves, every second to a P position, traversing the F-glider, illustrated in Figure 14. Using notation as in Corollary 5.3, we have the following result.

Corollary 5.4. *Let L, R and C denote finite binary strings. It is algorithmically undecidable whether, for a finite path of consecutive moves in a rule 110 game with an $\underline{L}C\underline{R}$ ending condition, every second position is P.*

Proof. By [C04], it is undecidable wether the time-wise pattern

$$\text{“110101010111111”} \tag{5.1}$$

ever appears in the update of the rule 110 CA given an $\underline{L}C\underline{R}$ initial condition. This pattern is produced in the interaction of the A- and C-gliders as the F-glider is created. Now, what is required is to describe a move path such that every second position is P and such that the pattern of the underlying CA-updates are equivalent to (5.1). We begin by showing that if every second position is P as indicated in the move path in Figure 14, then the pattern in (5.1) will appear in the 0th column in cell 1 to 15. The path of moves is $(1,16,1) \to (1,15,1) \to (2,13,2) \to (1,12,1) \to (2,9,3) \to (0,8,1) \to (0,7,1) \to (0,6,1) \to (0,5,1) \to (0,4,1) \to (0,3,1) \to (0,2,1) \to (1,0,2)$, where the origin is shifted as to simplify our description.

Starting from below, we find the following two moves: $(0,3,1) \to (0,2,1) \to (1,0,2)$. Now, translating (5.1) from left to right, these moves force the pattern “110”. Let us explain why this is so. The position $(0,2,1)$ is N, but has to cover a “1”, a dark (red) cell, since otherwise $(0,3,1)$ cannot be a P-position, because any P-position has to cover a “0” and simultaneously being supported by a “1”. Now, it remains to explain why the cell $(0,1)$

must cover a "1". Suppose it covers a "0". Then, by $(1, 0, 2)$ being P and by the updates of the CA, the cell $(1, 2)$ also contains a "0". But then, since $(0, 2)$ contains a "1", $(0, 3)$ must also, which contradicts its status as P. Hence the three leftmost binary digits are correct.

Now, the pattern "11010101" follows by a similar argument, corresponding to the 7 least positions in our move path. But for the following two binary digits "0" and "1" corresponding to the cells $(0, 8)$ and $(0, 9)$ respectively we need another argument. They are forced, because the position $(2, 9, 3)$ is P. Namely, it is immediate that the 3 cells $(0, 9), (1, 9), (2, 9)$ contain "0"s. But then cell $(0, 8)$ cannot contain a "0", again, since $(2, 9, 3)$ is P.

It remains to verify the last sequence of six "1"s in the binary bit string, starting with cells $(0, 10)$ to $(0, 12)$, which all must be "1"s since both $(2, 9, 3)$ and $(2, 13, 2)$ are P. Namely, together they force that each one of the cells $(0, 12), (1, 12), (2, 12), (1, 11), (0, 10)$ contains a "1". But then the updates of the CA produces a "1" also in cell $(0, 11)$.

Suppose now, for a contradiction, that cell $(0, 13)$ contains a "0". Then, so does cell $(2, 15)$, since $(2, 13, 2)$ is P. But $(1, 16, 1)$ is a P-position. Hence cell $(1, 15)$ contains a "1" and $(1, 16)$ contains a "0", which contradicts our assumption. Together with the update rules of the CA, this also implies that both $(0, 15)$ and $(0, 14)$ contain "1"s.

The other direction is immediate, since the binary string in (5.1) occurs if and only if the F-glider is produced by an interaction of the A- and C-gliders, which then also produces the rest of the patterns in Figure 14 as indicated. □

Returning to Lemma 4.5, and the first paragraph of Section 5, it remains an open question whether convergence of rule 110 games is decidable given two $\underline{L}C\underline{R}$ ending conditions. Via private communication with Matthew Cook we understand that such results do not follow from the methods used in [C04, C08]. Also, to our best knowledge, the problems of decidability discussed in this paper remain open for the CA($\underline{L}C\underline{R}, \Gamma, \gamma$) and corresponding game families for other combinations of γ and Γ than $\gamma \in \{0, 1\}$ and $\Gamma = 0$.

6. Discussion

The purpose of this paper has been to emulate well known cellular automata via impartial games following the normal play convention. On the one hand, there is an 'unintelligent system', in fact sometimes called a *zero player game*, with a very simple update function, which takes into account only the most recent history. On the other hand there are two combatants who play intelligently in the attempt of being the first to reach a given goal; and where arbitrarily large moves are allowed. In spite the apparent big differences between the games and the CA, we have showed that the patterns produced by the respective systems correspond precisely and hence they are equivalent in many respects. Moreover, since our rule 110 game emulate the rule 110 cellular automaton, we have demonstrated how the undecidability results from [C04, C08] transfer to our settings.

Other CA, for example from [W02], may have interesting interpretations as combinatorial games. A standard variation of a combinatorial game is to, at each stage of game, allow the previous player to block off exactly one of the next player's options [L11]. When a move is carried out, any blocking maneuver is forgotten. Suppose that we apply this blocking maneuver to the triangle-placing game. Is it possible to describe the P-positions for this game directly from the updates of some one-dimensional CA? In general, one can think of other rules for placing the triangles and/or using other types of triangles [L]. When do they correspond to one-dimensional CA via their outcomes?

A *disjunctive sum* of games consist of several components played simultaneously, but where it is legal to move in only one component at each stage of game, the winner being the player who makes the last move in the last component; [BCG04]. Note here, that the complexity of optimal play within each component is increased, since a sequence of moves in a specific component is not necessarily alternating between the two players. But, via the so-called *mex-algorithm*—which computes a non-negative integer called the *nim-value* for a normal play impartial game—the famous *Sprague-Grundy theory* provides a simple formula (*nim-sum*) to compute the nim-value for a *disjunctive sum* of a finite number of normal play impartial games. (The nim-value is 0 if and only if the second player to move wins.) Can the nim-values, of say our rule 60 and rule 110 games, be interpreted via the updates of some (non-binary) one-dimensional cellular automaton?

For a *partizan* [BCG04] variant of our games we suggest to color the terminal bit-string in a red-blue pattern. The players, "Red" and "Blue", play as usual, but Blue can place an IRT with its base covering the terminal bit-string if and only if it covers only blue cells, whereas Red's final IRT must cover only red cells at the terminal level. Another partizan variation is to let the players move from different triangle supports, but where the terminal bit-string is the same for both players. For example, let Red move from supports as in rule 60 and Blue from those in rule 110. (In a sum of games this is not the same as saying that Red places rule 110 triangles, whereas Blue places rule 60 ditto; it is better to simply let the players place IRTs.) There are four outcome classes for partizan games. Is there any correspondence of them to updates of some one-dimensional cellular automaton? The intuition, of course, is that Blue is favored in this sample *rule 60/110 game* and that it could at least be experimentally verified (via canonical game values) for small game boards.

Acknowledgments. I thank David Wahlstedt and Matthew Cook for inspiring conversations on undecidability and Mike Weimerskirch for many good suggestions. I am also grateful to the anonymous referees, whose comments helped to improve the paper.

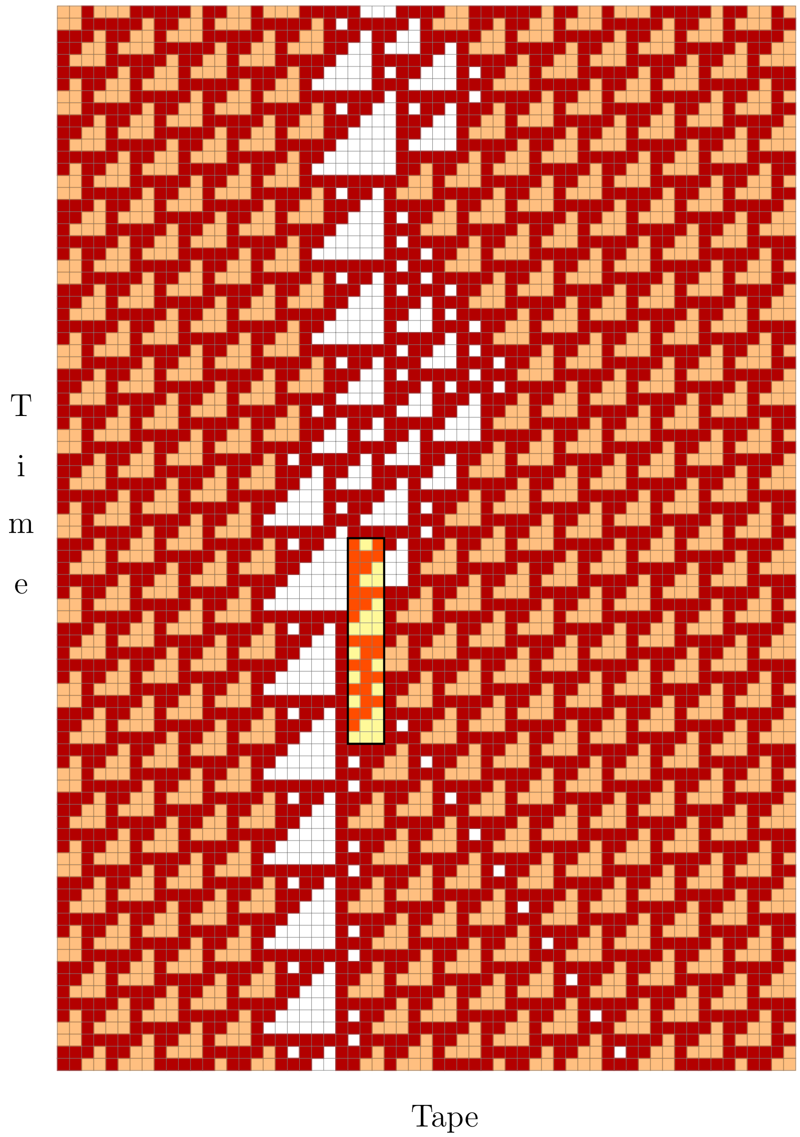

Figure 13: A collision of a C-glider (running North) and an A-glider (running North West), in background ether, is captured in the central frame of 3×17 cells. The collision creates the more complex F-glider, of which a good period is displayed (running North North East). The bit string "1101010101111111", which is contained in the central frame, is also displayed in Figure 14. We have indicated the "0"-cells of the gliders in white (and yellow). The other "0"-cells (in orange) belong to the 14 cell spatially periodic background ether.

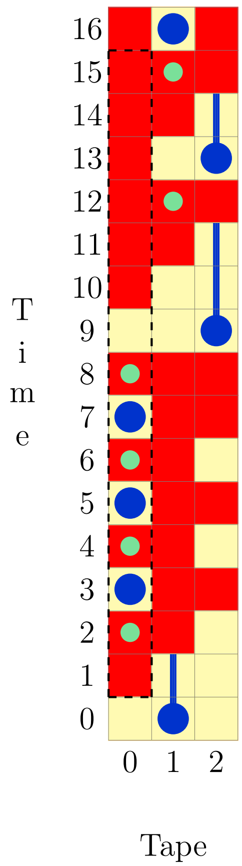

Figure 14: The pattern "110101010111111" is contained within the dashed frame—the content of the first to the 15th cell in the 0th column—as it appears in the creation of the F-glider by the collision of the A- and C-gliders in Figure 13. A corresponding alternating path of P- and N-positions, with notation as in the triangle-placing game, is also shown: $(1, 16, 1) \to (1, 15, 1) \to (2, 13, 2) \to (1, 12, 1) \to (2, 9, 3) \to (0, 8, 1) \to (0, 7, 1) \to (0, 6, 1) \to (0, 5, 1) \to (0, 4, 1) \to (0, 3, 1) \to (0, 2, 1) \to (1, 0, 2)$.

References

[BCG04] E. R. Berlekamp, J. H. Conway, R. K. Guy, *Winning ways*, 1-2 Academic Press, London (1982). Second edition, 1-4. A. K. Peters, Wellesley/MA (2001/03/03/04).

[C98] M. Cook. Computational universality in elementary cellular automata. In David Griffeath and Cristopher Moore, editors, *New Constructions in Cellular Automata*, Oxford, (2000). Oxford Univ. Press. Proceedings of the Constructive Cellular Automata Theory conference, Santa Fe Institute, Santa Fe, New Mexico, (1998) 14–17.

[C04] M. Cook, Universality in Elementary Cellular Automata, *Complex Systems*, 15 (2004) 1–40.

[C08] M. Cook, A Concrete View of Rule 110 Computation, *The Complexity of Simple Programs 2008, EPTCS 1*, (2009) 31–51.

[DH09] E. D. Demaine and R. A. Hearn, Playing Games with Algorithms: Algorithmic Combinatorial Game Theory, *Games of No Chance 3, MSRI Publications*, edited by Michael H. Albert and Richard J. Nowakowski, Cambridge Univ. Press., 56 (2009) 3–56.

[DH] R. A. Hearn and Erik D. Demaine, *Games, Puzzles, and Computation*, A. K. Peters, (2009).

[F12] A. Fink, Lattice games without rational strategies *J. Combin. Theory*, Series A 119 (2012) 450–459.

[F02] A. S. Fraenkel, Two-player games on cellular automata, in R. J. Nowakowski, ed., *More Games of No Chance, Proc. MSRI Workshop Combinatorial Games* (Berkeley CA, 2000), Cambridge Univ. Press, (2002).

[G66] S. W. Golomb, A mathematical investigation of games of "take-away". *J. Combin. Theory* 1 (1966) 443–458.

[HU79] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Reading, MA: Addison-Wesley, (1979).

[L09] U. Larsson, 2-pile Nim with a Restricted Number of Move-size Imitations, *Integers* 9 (2009), Paper G4, 671–690.

[L11] U. Larsson, Blocking Wythoff Nim, *Electron. J. Combin.* 18 (2011) no. 1, Paper 120, 18 pp.

[L12] U. Larsson, The ⋆-operator and invariant subtraction games, 422 *Theoret. Comput. Sci.* (2012) 52–58.

[L] U. Larsson, Self-similarities in cellular automata and combinatorial games, in preparation.

[LW] U. Larsson, J. Wästlund, From heaps of matches to undecidability of games, http://arxiv.org/pdf/1202.0664.pdf.

[N66] J. von Neumann, Theory of Self-Reproducing Automata (A. W. Burks, ed.), *Champaign, IL: University of Illinois Press*, (1966).

[S70] A. J. Schwenk, "Take-Away Games", *Fibonacci Quart.* 8 (1970) 225–234.

[W84a] S. Wolfram, Universality and Complexity in Cellular Automata, *Physica*, 10D, (1984) 1–35.

[W84b] S. Wolfram, Cellular Automata as Models of Complexity, *Nature*, 311(4), (1984) 419–424.

[W84c] S. Wolfram, *Computation Theory of Cellular Automata*, Communications in Mathematical Physics, 96(1), (1984) 15–57.

[W86] S. Wolfram, *Theory and Application of Cellular Automata*, Singapore: World Scientific, (1986).

[W02] S. Wolfram, *A New Kind of Science*, Champaign, IL: Wolfram Media, Inc., (2002).

[Z96] M. Zieve, Take-Away Games, *Games of No Chance, MSRI Publications*, 29, (1996) 351–361.