Contributions to Discrete Mathematics

Volume 20, Number 2, Pages 58–97
ISSN 1715-0868

ON EMULATIONAL EQUIVALENCE OF IMPARTIAL GAMES AND THE GAME HACKENFORB

BOJAN BAŠIĆ, NIKOLA MILOSAVLJEVIĆ, AND DANIJELA POPOVIĆ

Abstract. We introduce a variant of the game Hackenbush, called *Hackenforb*. It is a class of games, each of which is determined by two parameters: a given graph, and a given set of connected graphs (called *forbidden graphs*). The significance of this game within the realm of impartial combinatorial games is reflected in the fact that, as we show in this article, various known combinatorial games, such as Nim, Subtraction game, Notakto, Treblecross, Chomp, are emulationally equivalent to an instance of Hackenforb (an *emulational equivalence* of two games is a concept stronger than Grundy-equivalence, but weaker than the isomorphism between games' structures; our belief is that this version of equivalence is what really captures the core of the intuitive perception of what it means for two games to be "basically the same game"). At the end of our article, we show that Hackenforb is, unfortunately, not "almighty," that is, we describe a game that is not emulationally equivalent to an instance of Hackenforb.

1. Introduction

Impartial games on graphs where players choose vertices and/or edges to be thence removed are a popular research topic; see [12, 9, 6, 3, 1], as well as [7, 8]. In particular, in the last two cited articles, in each move a player is entitled to choose one edge, which is then removed (possibly together with some more edges/vertices, as specified by the rules of a particular game). Arguably the most well-known game that fits this description is the so-called Hackenbush (the impartial form of it), invented in the early 1970s by Conway [5, 4]. In this game, a graph is given whose one or more vertices are denoted as "the ground" (usually represented as a horizontal line) and

This work is licensed under a Creative Commons "Attribution-NoDerivatives 4.0 International" license.

Received by the editors November 4, 2021, and in revised form October 12, 2023.
2000 *Mathematics Subject Classification.* 05C57, 91A43, 91A46.
Key words and phrases. Hackenbush, Hackenforb, emulational equivalence, Nim, Subtraction game, Notakto, Treblecross, Chomp, nimber, Grundy value.
The first two authors were supported by the Ministry of Education, Science and Technological Development of Serbia (grant no. 451-03-47/2023-01/200125, respectively 451-03-68/2020-14/200124), and the third author was supported by the same Ministry through Mathematical Institute of the Serbian Academy of Sciences and Arts.

every vertex is connected to the ground (not necessarily directly). Players alternately remove one edge at a time, and if after removing an edge a connected component appears that is disconnected from the ground, that whole component consequently gets removed in the same move. The game ends when there are no more possible moves.

In the present article, we introduce a variant of this game, called *Hackenforb.* The game is played on a graph given in advance, and the rules are similar, with the only difference that there is no notion of "ground" in Hackenforb, but instead, if after a move a connected component appears that belongs to the set of forbidden connected components also given in advance, then that component gets removed in the same move (or any such component, if there are more than one of them). Coincidentally, the game *Disconnect-it* from the already mentioned article [7] is a very special case of Hackenforb (when the initial graph is K_n and the set of forbidden connected components is the set of all connected subgraphs of K_n of order less than n).

Let us explain the significance of the game Hackenforb within the realm of impartial combinatorial games. For that, we first need to introduce the concept of emulational equivalence between two games. To say as a kind of "advertisement," our aim is to define this notion in such a way that two games are *emulationally equivalent* if they are basically the same game (but possibly expressed in different languages). Recall that, in combinatorial game theory, when the *normal* play is concerned (the player who is unable to move loses the game), two games are considered equivalent if their corresponding (so-called) *nimbers* (or *nim-values*, *Grundy values*) are the same. The very rich theory based on this notion is an irreplaceable tool when questions of the winner and the optimal strategy are concerned. However, this theory does not really tell much about the structural (dis)similarity between two games. On the other hand, we could say that two games are "the same" if their structures (the set of all possible positions and all possible moves between them) are isomorphic. However, this is too restrictive, and happens very rarely. We believe that what really captures the core of the intuitive perception of what it means for two games to be "basically the same game" is the notion of emulational equivalence. It bears a resemblance to the isomorphism between games' structures, but it is slackened by the fact that positions that are "essentially the same" are not treated as distinct. What do we mean by "essentially the same"? For example, if two positions have exactly the same sets of possible moves, they can be perceived as essentially the same (in particular, all the ending positions are essentially the same). As another example, positions that are mapped to each other under some kind of symmetry can be perceived as essentially the same. But this is still not all; some less obvious examples will be presented later. In general, we say that some positions are "essentially the same" if their sets of possible moves are "essentially the same" (this is, of course, a very loose description; here we are aiming only for a rough sketch to help the reader's intuition, while a rigorous definition will be given in Subsection 2.1, and, of course, will not be of a circular nature).

As we shall see in Section 3, many impartial games can be expressed as (that is: are emulationally equivalent to) an instance of Hackenforb (recall that an instance of Hackenforb is determined by a given graph that represents the initial position, as well as a given set of forbidden connected components). Let us summarize the content of that section.

- In Subsection 3.1 we show that each instance of the game Nim is emulationally equivalent to an instance of Hackenforb. Given the major role the game Nim has within combinatorial game theory, this is a natural first step in studying the "emulation potential" of the game Hackenforb.
- In Subsection 3.2 we consider a variant of Nim sometimes called the Subtraction game: a heap of n coins is given, and in each move a player takes a positive number of coins not greater than a bound k given before the start of the game. We show that this game (for any n and k) is emulationally equivalent to an instance of Hackenforb.
- In Subsection 3.3 we consider a not-so-well-known game called Notakto (see, e.g., [11]), which can be simply described as an impartial version of tic-tac-toe. The main contribution of Subsection 3.3 is maybe not the result itself, but the fact that our approach works (with minimal modifications) for more-or-less any game where players are entitled to claim a field of a playing board, striving to achieve or avoid a particular pattern. To mention just one more example, Notakto's one-dimensional (and probably better-known) cousin Treblecross [2, p. 94] [10, p. 281] is also emulationally equivalent to an instance of Hackenforb, which can be proved by a straightforward adaptation of our proof of Theorem 3.12.
- Subsection 3.4 is the most technical part of the article. Therein, we prove that the well-known game Chomp (first introduced by Schuh in [13]) is emulationally equivalent to an instance of Hackenforb. In order to prove this, we introduce an auxiliary game called Auxie, and then show that Chomp is emulationally equivalent to Auxie, as well as that Auxie is emulationally equivalent to an instance of Hackenforb. There is even a (maybe slim) chance that our game Auxie might turn out to be of independent interest in the future, since, given many still open questions on Chomp (see, e.g., [14]), we could at least hope that a game that is "basically the same" as Chomp but whose positions capture some additional peculiarities (in comparison to positions of Chomp) might provide some new insights on those hard problems.

Finally, in Section 4 we show that Hackenforb is not almighty, that is, we describe a game that is not emulationally equivalent to an instance of Hackenforb. We admit that, for a brief period of time, we were even hoping that it could be the case that *all* impartial games are emulationally equivalent to an instance of Hackenforb. The result of this section shatters that hope, but we still have no doubts that Hackenforb is a unifying tool capable of more than enough to vindicate its existence.

2. Preliminaries

We first need to formally state what we mean by a game for the purpose of this article.

Definition 2.1. *A* game *is a triple* (P, f, p_0)*, where* P *is any finite set,* f *a function* $f : P \to \mathcal{P}(P)$*, and* p_0 *an element from* P*. In this notation,* P *represents the set of all possible game positions,* p_0 *is the starting position, and* f *maps a position* p *to all the positions that can be reached from* p *in one move. We additionally require that there do not exist* $p_1, p_2, \ldots, p_k \in P$ *such that* $p_2 \in f(p_1), p_3 \in f(p_2), \ldots, p_k \in f(p_{k-1}), p_1 \in f(p_k)$*, and that* p_0 *is the unique element from* P *with the property that it does not belong to any* $f(p)$ *for* $p \in P$*.*

The game ends (naturally) when there are no more possible moves, but one could object that the definition does not state who wins the game. Since it is only the game structure that interests us, this was intentionally left out from the definition.

We now proceed to define the concept of emulational equivalence between games, and show some basic assertions about it.

2.1. **Emulational equivalence.**

Definition 2.2. *For a game* $\mathcal{G} = (P, f, p_0)$ *and a partition* $\mathcal{M}$ *of* P *we say that* $\mathcal{M}$ *is a* congruential partition of $\mathcal{G}$ *iff*

$$(\forall A, B \in \mathcal{M})((\exists a \in A)(\exists b \in B)(b \in f(a)) \Rightarrow (\forall x \in A)(\exists y \in B)(y \in f(x))). \tag{2.1}$$

For such sets $A, B \in \mathcal{M}$ *we shall write* $B \in f(A)$*.*

In other words: if there exists a move from somewhere in A to anywhere in B, then, *wherever* we find ourselves in the class A, it must be possible to move to somewhere in the class B. On an intuitive level, each class in a congruential partition gathers together positions that can be treated as "essentially the same."

Note that each game has at least one congruential partition: namely, $\mathcal{M} = \{\{p\} : p \in P\}$. This partition will be called *trivial partition.*

As usual, if $\mathcal{M}$ is a partition of a set P, and ρ is the induced equivalence relation, then for $a \in P$ we write

$$[a]_\rho = [a]_\mathcal{M} = \{x \in P : x \ \rho \ a\}.$$

Definition 2.3. *We say that games* $\mathcal{G}_1 = (P_1, f_1, a_1)$ *and* $\mathcal{G}_2 = (P_2, f_2, a_2)$ *are* emulationally equivalent *if there exist congruential partitions* $\mathcal{M}_1$ *and*

$\mathcal{M}_2$ of $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively, and a bijection $F : \mathcal{M}_1 \to \mathcal{M}_2$ such that $F([a_1]_{\mathcal{M}_1}) = [a_2]_{\mathcal{M}_2}$ and

$$(\forall A, B \in \mathcal{M}_1)(B \in f_1(A) \Leftrightarrow F(B) \in f_2(F(A))). \tag{2.2}$$

We shall now give one basic example of a congruential partition and an emulational equivalence. We assume the reader is familiar with the game of Nim and its rules (if not, see Subsection 3.1).

Example. *We consider the game of Nim with two heaps of* 3 *and* 2 *coins, respectively. Formally, each position can be considered either as an ordered pair or as a two-element multiset. Figure 1 left shows the game graph in the first case, while Figure 1 right shows the game graph in the second case. Although these two game graphs are different, we are clearly speaking about essentially the same game.*

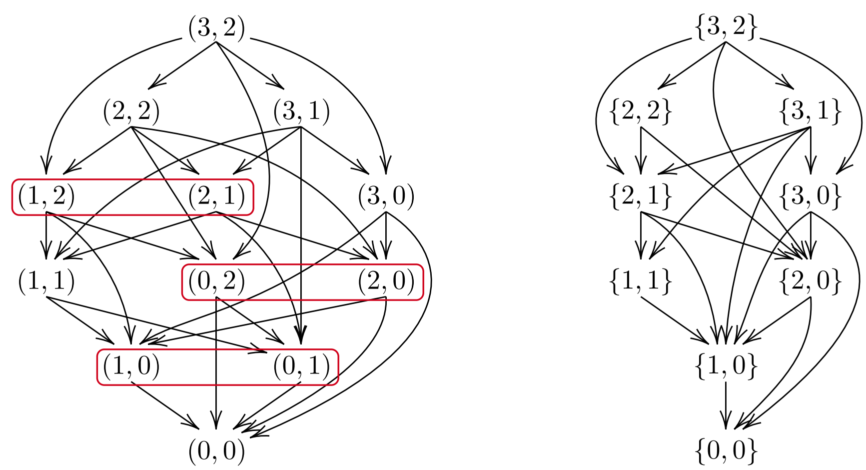

FIGURE 1. Game graphs of two possibilities of a formal definition of Nim.

Figure 1 left also shows one nontrivial congruential partition: the positions (m, n) and (n, m) for $m, n \in \{0, 1, 2\}$, $m \neq n$, are in the same class (such classes are marked by the red boxes), while all the other positions are alone in their classes. The bijection of this partition of the game on the left and the trivial partition of the game on the right is now obvious, and it is easy to check that this bijection satisfies (2.2).

The previous example could leave an incorrect impression that emulational equivalence always originates from some kind of symmetry. Here is a different example.

Example. *In* Grundy's game [2, p. 96], *a move consists of dividing a heap of coins into two non-equal heaps. In Figure 2 we show the game graph of the game played on a heap of size* 7*. Grouped together by red boxes are positions that are in the same class in one congruential partition of this game (without* *any visible underlying symmetry; although it is not hard to see that positions that differ only in heaps of sizes* 1 *or* 2 *are essentially the same, this is still not enough to explain the third row from the bottom).*

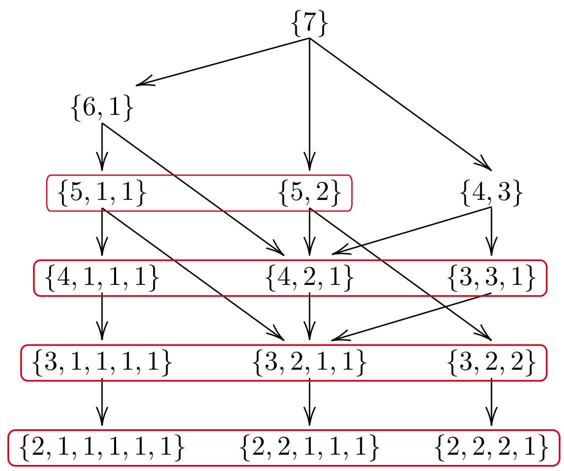

FIGURE 2. Game graph of Grundy's game played on a heap of size 7.

We shall now briefly recall some basic notions on equivalence relations. If ρ_1 and ρ_2 are two equivalence relations on P and $\rho_1 \subseteq \rho_2$, then we shall write $\mathcal{M}_1 \leqslant \mathcal{M}_2$ for their corresponding quotient sets (partitions of P). (For each a, $a \in P$, we then have $[a]_{\mathcal{M}_1} \subseteq [a]_{\mathcal{M}_2}$, and moreover, each class in $\mathcal{M}_2$ is a union of some classes from $\mathcal{M}_1$.) Also, for $A \in \mathcal{M}_1$, we shall write $[A]_{\mathcal{M}_2}$ to denote $[a]_{\mathcal{M}_2}$ where $a \in A$ (note that, because of $\mathcal{M}_1 \leqslant \mathcal{M}_2$, the choice of a does not matter). Finally, if ρ is a relation on P (not necessarily an equivalence relation), then $\mathrm{tr}(\rho)$ will denote the transitive closure of ρ (defined as the smallest relation on P that contains ρ and is transitive).

Our first aim is to prove that the emulational equivalence is an equivalence relation on the class of all games. For that, we shall need two preparatory lemmas.

Lemma 2.4. *Given a game $\mathcal{G} = (P, f, p_0)$, let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two congruential partitions of it. If ρ_1 and ρ_2 are the corresponding equivalence relations on P, and $\rho = \mathrm{tr}(\rho_1 \cup \rho_2)$, then the quotient set $\mathcal{M}$ of ρ is also a congruential partition of the game $\mathcal{G}$.*

Proof. First note that, since ρ_1 and ρ_2 are reflexive and symmetric, then ρ is also; therefore, ρ is indeed an equivalence relation and $\mathcal{M}$ is well-defined. Let us now prove (2.1).

Let $A, B \in \mathcal{M}$, let $a \in A = [a]_{\mathcal{M}}$, $b \in B = [b]_{\mathcal{M}}$, and let $b \in f(a)$. Let $x \in [a]_{\mathcal{M}}$. We need to prove that there is y, $y \in [b]_{\mathcal{M}}$, such that $y \in f(x)$.

Since $\rho = \mathrm{tr}(\rho_1 \cup \rho_2)$, it follows that there exist $a_1, \dots, a_n \in P$ such that

$$a\ (\rho_1 \cup \rho_2)\ a_1\ (\rho_1 \cup \rho_2)\ a_2 \dots a_n\ (\rho_1 \cup \rho_2)\ x.$$

Since $a\ (\rho_1 \cup \rho_2)\ a_1$, we have $a\ \rho_1\ a_1$ or $a\ \rho_2\ a_1$. Assume first $a\ \rho_1\ a_1$. Then $a_1 \in [a]_{\mathcal{M}_1}$. Since $b \in f(a)$, by (2.1) (recalling that $\mathcal{M}_1$ is a congruential partition) there exists $b_1 \in P$ such that $b_1 \in [b]_{\mathcal{M}_1}$ and $b_1 \in f(a_1)$. Since $\rho_1 \subseteq \rho_1 \cup \rho_2 \subseteq \rho$, we have $\mathcal{M}_1 \leqslant \mathcal{M}$ and $b_1 \in [b]_{\mathcal{M}}$. In a similar way, if $a\ \rho_2\ a_1$, we can find $b_1 \in [b]_{\mathcal{M}_2} \subseteq [b]_{\mathcal{M}}$ and $b_1 \in f(a_1)$.

Analogously, there exist $b_2, \ldots, b_n \in P$ such that $b_i \in f(a_i)$ and $b_i \in [b_{i-1}]_{\mathcal{M}} = [b]_{\mathcal{M}}$, for $i \in \{2, \ldots, n\}$. At last, $a_n\ (\rho_1 \cup \rho_2)\ x$ and $b_n \in f(a_n)$ imply that there exists $y \in [b_n]_{\mathcal{M}} = [b]_{\mathcal{M}}$ such that $y \in f(x)$, which was to be proved. □

Lemma 2.5. *Let $\mathcal{G}_1 = (P_1, f_1, a_1)$ and $\mathcal{G}_2 = (P_2, f_2, a_2)$ be two emulationally equivalent games, with congruential partitions $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively, and let F be a bijection $F : \mathcal{M}_1 \to \mathcal{M}_2$ that satisfies (2.2) and $F([a_1]_{\mathcal{M}_1}) = [a_2]_{\mathcal{M}_2}$. Let $\mathcal{M}$ be a congruential partition of $\mathcal{G}_1$ such that $\mathcal{M}_1 \leqslant \mathcal{M}$, and let us define a relation ρ' on P_2 by:*

$$(\forall x, y \in P_2)(x\ \rho'\ y \Leftrightarrow [F^{-1}([x]_{\mathcal{M}_2})]_{\mathcal{M}} = [F^{-1}([y]_{\mathcal{M}_2})]_{\mathcal{M}}). \tag{2.3}$$

Then ρ' is an equivalence relation, and its quotient set $\mathcal{M}'$ is a congruential partition of $\mathcal{G}_2$. Moreover, we have $\mathcal{M}_2 \leqslant \mathcal{M}'$, and there is a bijection $F' : \mathcal{M} \to \mathcal{M}'$ that satisfies the condition (2.2) and $F'([a_1]_{\mathcal{M}}) = [a_2]_{\mathcal{M}'}$.

Proof. Since $F : \mathcal{M}_1 \to \mathcal{M}_2$ is a bijection, we have $F^{-1}([x]_{\mathcal{M}_2}) \in \mathcal{M}_1$. Since $\mathcal{M}_1 \leqslant \mathcal{M}$, we have that $[F^{-1}([x]_{\mathcal{M}_2})]_{\mathcal{M}}$ is well-defined for all $x \in P_2$. Therefore, ρ' is an equivalence relation since the equality is an equivalence relation. Also, if $[x]_{\mathcal{M}_2} = [y]_{\mathcal{M}_2}$, from (2.3) follows $x\ \rho'\ y$, that is, $[x]_{\mathcal{M}'} = [y]_{\mathcal{M}'}$. Therefore, $\mathcal{M}_2 \leqslant \mathcal{M}'$.

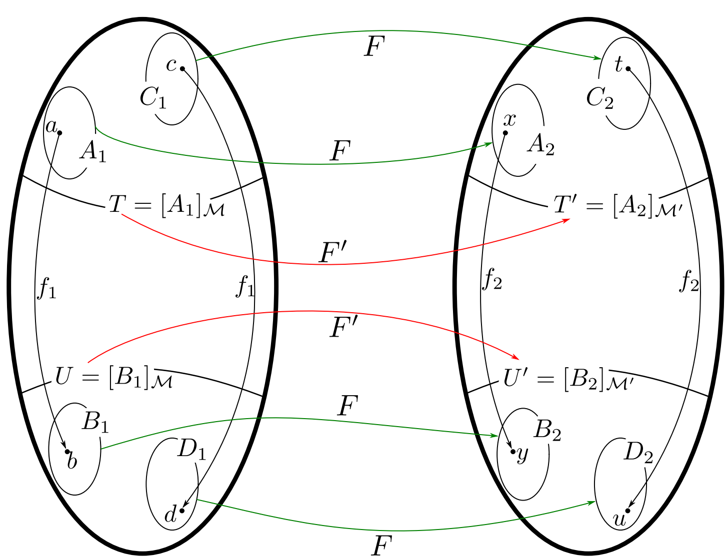

FIGURE 3. An illustration for the proof of Lemma 2.5.

Let us now prove that $\mathcal{M}'$ is a congruential partition, that is, that it satisfies the condition (2.1). Let $T', U' \in \mathcal{M}'$ and $x, t \in T'$, $y \in U'$ such that $y \in f_2(x)$. We want to prove that there exists $u \in U'$ such that $u \in f_2(t)$. Let $A_2 = [x]_{\mathcal{M}_2}$, $B_2 = [y]_{\mathcal{M}_2}$, $C_2 = [t]_{\mathcal{M}_2}$, and let $A_1 = F^{-1}(A_2)$, $B_1 = F^{-1}(B_2)$, $C_1 = F^{-1}(C_2)$. (See Figure 3. Note that this figure also contains some unnecessities, which is because the same figure will again be used in the later part of the proof. We believe, though, that this will not cause any confusion to the reader.) Since $B_2 \in f_2(A_2)$, by (2.2) we conclude $B_1 \in f_1(A_1)$, which further implies

$$[B_1]_{\mathcal{M}} \in f_1([A_1]_{\mathcal{M}}) \tag{2.4}$$

(because $\mathcal{M}$ is a congruential partition and $\mathcal{M}_1 \leqslant \mathcal{M}$). Since $x\ \rho'\ t$ (by the choice of x, t), we get $[F^{-1}([x]_{\mathcal{M}_2})]_{\mathcal{M}} = [F^{-1}([t]_{\mathcal{M}_2})]_{\mathcal{M}}$ (by (2.3)), and thus

$$\begin{aligned}[A_1]_{\mathcal{M}} &= [F^{-1}(A_2)]_{\mathcal{M}}\\ &= [F^{-1}([x]_{\mathcal{M}_2})]_{\mathcal{M}}\\ &= [F^{-1}([t]_{\mathcal{M}_2})]_{\mathcal{M}}\\ &= [F^{-1}(C_2)]_{\mathcal{M}}\\ &= [C_1]_{\mathcal{M}}.\end{aligned}$$

Together with (2.4), this implies that for any $c \in C_1$ there exists $d \in [B_1]_{\mathcal{M}}$ such that $d \in f_1(c)$. Choose any such c, and let $D_1 = [d]_{\mathcal{M}_1}$; we then have $D_1 \subseteq [D_1]_{\mathcal{M}} = [B_1]_{\mathcal{M}}$. Let $F(D_1) = D_2 \in \mathcal{M}_2$. Note that $d \in f_1(c)$ implies $D_1 \in f_1(C_1)$ (since $\mathcal{M}_1$ is a congruential partition) and then $D_2 \in f_2(C_2)$ (the condition (2.2) for F). Since $t \in C_2$, there exists $u \in D_2$ such that $u \in f_2(t)$. Now from

$$\begin{aligned}[F^{-1}([u]_{\mathcal{M}_2})]_{\mathcal{M}} &= [F^{-1}(D_2)]_{\mathcal{M}}\\ &= [D_1]_{\mathcal{M}}\\ &= [B_1]_{\mathcal{M}}\\ &= [F^{-1}(B_2)]_{\mathcal{M}}\\ &= [F^{-1}([y]_{\mathcal{M}_2})]_{\mathcal{M}}\end{aligned}$$

and (2.3) follows $u\ \rho'\ y$, that is, $u \in U'$, which concludes the proof that $\mathcal{M}'$ is a congruential partition of the game $\mathcal{G}_2$.

We are left to find a bijection $F' : \mathcal{M} \to \mathcal{M}'$ that satisfies (2.2) and $F'([a_1]_{\mathcal{M}}) = [a_2]_{\mathcal{M}'}$. Let A_1 be an arbitrary class from $\mathcal{M}_1$ (that is, A_1 is not the specific class from the previous paragraph anymore, and we use the same letter because that way Figure 3 will be of use to follow this part of the proof, too, as we have already mentioned). We define F' by

$$F'([A_1]_{\mathcal{M}}) = [F(A_1)]_{\mathcal{M}'}.$$

Note that for each class $T \in \mathcal{M}$ there exists a class $A_1 \in \mathcal{M}_1$ such that $T = [A_1]_{\mathcal{M}}$ (because of $\mathcal{M}_1 \leqslant \mathcal{M}$), while the right-hand side makes sense because of $F(A_1) \in \mathcal{M}_2$ and $\mathcal{M}_2 \leqslant \mathcal{M}'$. Also, F is well-defined, that is: whenever $[A_1]_{\mathcal{M}} = [C_1]_{\mathcal{M}}$ (for $A_1, C_1 \in \mathcal{M}_1$), then also $[F(A_1)]_{\mathcal{M}'} = [F(C_1)]_{\mathcal{M}'}$, which follows by the definition of $\mathcal{M}'$. The function F' is injective since, if $[F(A_1)]_{\mathcal{M}'} = [F(C_1)]_{\mathcal{M}'}$ for some $A_1, C_1 \in \mathcal{M}_1$, then (2.3) implies $[A_1]_{\mathcal{M}} = [F^{-1}(F(A_1))]_{\mathcal{M}} = [F^{-1}(F(C_1))]_{\mathcal{M}} = [C_1]_{\mathcal{M}}$. Now we want to prove that F' is surjective. For any class $T' \in \mathcal{M}'$, there exists $A_2 \in \mathcal{M}_2$ such that $T' = [A_2]_{\mathcal{M}'}$. Let $A_1 = F^{-1}(A_2)$; then $F'([A_1]_{\mathcal{M}}) = [F(A_1)]_{\mathcal{M}'} = [A_2]_{\mathcal{M}'} = T'$, and F' is indeed surjective.

Altogether, we have that F' is a bijection. Also,

$$F'([a_1]_{\mathcal{M}}) = F'([[a_1]_{\mathcal{M}_1}]_{\mathcal{M}}) = [F([a_1]_{\mathcal{M}_1})]_{\mathcal{M}'} = [[a_2]_{\mathcal{M}_2}]_{\mathcal{M}'} = [a_2]_{\mathcal{M}'}.$$

In order to finish the proof, we have to show that F' satisfies the condition (2.2).

($\Rightarrow$): Suppose that for some classes $T, U \in \mathcal{M}$ we have $U \in f_1(T)$. We want to prove $F'(U) \in f_2(F'(T))$. Let $a \in T$ and $A_1 = [a]_{\mathcal{M}_1}$; note that $[A_1]_{\mathcal{M}} = T$. From $U \in f_1(T)$ follows that there exists $b \in U$ such that $b \in f_1(a)$. Let $B_1 = [b]_{\mathcal{M}_1}$; note that $[B_1]_{\mathcal{M}} = U$. Since $B_1 \in f_1(A_1)$, the condition (2.2) for F gives $F(B_1) \in f_2(F(A_1))$. Since $\mathcal{M}'$ is a congruential partition, we conclude

$$\begin{aligned} F'(U) &= F'([B_1]_{\mathcal{M}}) \\ &= [F(B_1)]_{\mathcal{M}'} \\ &\in f_2([F(A_1)]_{\mathcal{M}'}) \\ &= f_2(F'([A_1]_{\mathcal{M}})) \\ &= f_2(F'(T)), \end{aligned}$$

which was to be proved.

($\Leftarrow$): Suppose that for some classes $T', U' \in \mathcal{M}'$ we have $U' \in f_2(T')$. Let $U = (F')^{-1}(U')$ and $T = (F')^{-1}(T')$. We want to prove $U \in f_1(T)$. Let $x \in T'$ and $A_2 = [x]_{\mathcal{M}_2}$; note that $[A_2]_{\mathcal{M}'} = T'$. The assumption $U' \in f_2(T')$ means that there exists $y \in U'$ such that $y \in f_2(x)$. Let $B_2 = [y]_{\mathcal{M}_2}$; note that $[B_2]_{\mathcal{M}'} = U'$. Let $A_1 = F^{-1}(A_2)$ and $B_1 = F^{-1}(B_2)$. Since $B_2 \in f_2(A_2)$, the condition (2.2) for F gives

$$B_1 = F^{-1}(B_2) \in f_1(F^{-1}(A_2)) = f_1(A_1). \tag{2.5}$$

By the definition of F', we have:

$$T = (F')^{-1}(T') = (F')^{-1}([A_2]_{\mathcal{M}'}) = (F')^{-1}([F(A_1)]_{\mathcal{M}'}) = [A_1]_{\mathcal{M}};$$
$$U = (F')^{-1}(U') = (F')^{-1}([B_2]_{\mathcal{M}'}) = (F')^{-1}([F(B_1)]_{\mathcal{M}'}) = [B_1]_{\mathcal{M}}.$$

Together with (2.5), this gives $U \in f_1(T)$, which completes the proof. □

Theorem 2.6. *Emulational equivalence is a reflexive, symmetric and transitive relation.*

Proof. We first prove that it is reflexive. Let $\mathcal{G}_1 = (P_1, f_1, a_1)$. Let $\mathcal{M}$ be the trivial partition of $\mathcal{G}_1$. If F is the identity function on $\mathcal{M}$, then $F(\{a_1\}) = \{a_1\}$ and (2.2) is clearly fulfilled.

If $\mathcal{G}_1 = (P_1, f_1, a_1)$ and $\mathcal{G}_2 = (P_2, f_2, a_2)$ are two emulationally equivalent games, $\mathcal{M}_1$ and $\mathcal{M}_2$ the corresponding congruential partitions and F the bijection between partitions satisfying (2.2) and $F([a_1]_{\mathcal{M}_1}) = [a_2]_{\mathcal{M}_2}$, then the function F^{-1} witnesses that $\mathcal{G}_2$ and $\mathcal{G}_1$ are emulationally equivalent games.

We are left to prove the transitivity. Let $\mathcal{G}_1 = (P_1, f_1, a_1)$, $\mathcal{G}_2 = (P_2, f_2, a_2)$ and $\mathcal{G}_3 = (P_3, f_3, a_3)$ be three games where $\mathcal{G}_1$ and $\mathcal{G}_2$ are emulationally equivalent, and $\mathcal{G}_2$ and $\mathcal{G}_3$ are also emulationally equivalent. Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be congruential partitions of $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively, with a bijection $F_1 : \mathcal{M}_1 \to \mathcal{M}_2$ that satisfies the condition (2.2) and $F_1([a_1]_{\mathcal{M}_1}) = [a_2]_{\mathcal{M}_2}$. In a similar way, let $\mathcal{M}'_2$ and $\mathcal{M}_3$ be congruential partitions of $\mathcal{G}_2$ and $\mathcal{G}_3$, respectively, with a bijection $F_2 : \mathcal{M}'_2 \to \mathcal{M}_3$ that satisfies the condition (2.2) and $F_2([a_2]_{\mathcal{M}'_2}) = [a_3]_{\mathcal{M}_3}$. Let ρ_2 and ρ'_2 be the equivalence relations on P_2 whose quotient sets are $\mathcal{M}_2$ and $\mathcal{M}'_2$, respectively. Let $\rho = \mathrm{tr}(\rho_2 \cup \rho'_2)$, and let $\mathcal{M}$ be the quotient set of ρ. By Lemma 2.4, $\mathcal{M}$ is also a congruential partition of the game $\mathcal{G}_2$ and $\mathcal{M}_2, \mathcal{M}'_2 \leqslant \mathcal{M}$.

Since the function $F_1^{-1} : \mathcal{M}_2 \to \mathcal{M}_1$ is a bijection that satisfies the condition (2.2), $F_1^{-1}([a_2]_{\mathcal{M}_2}) = [a_1]_{\mathcal{M}_1}$, and $\mathcal{M}_2 \leqslant \mathcal{M}$, we can apply Lemma 2.5. There exists a congruential partition $\mathcal{M}'_1$ of $\mathcal{G}_1$ and a bijection $F'_1 : \mathcal{M} \to \mathcal{M}'_1$ that satisfies (2.2) and $F'_1([a_2]_{\mathcal{M}}) = [a_1]_{\mathcal{M}'_1}$. Then its inverse function $(F'_1)^{-1} : \mathcal{M}'_1 \to \mathcal{M}$ also witnesses emulational equivalence of the games $\mathcal{G}_1$ and $\mathcal{G}_2$. Similarly, since $F_2 : \mathcal{M}'_2 \to \mathcal{M}_3$ is a bijection that satisfies (2.2), $F_2([a_2]_{\mathcal{M}'_2}) = [a_3]_{\mathcal{M}_3}$ and $\mathcal{M}'_2 \leqslant \mathcal{M}$, by Lemma 2.5 there exists a congruential partition $\mathcal{M}'_3$ of $\mathcal{G}_3$ and a bijection $F'_2 : \mathcal{M} \to \mathcal{M}'_3$ that satisfies (2.2) and $F'_2([a_2]_{\mathcal{M}}) = [a_3]_{\mathcal{M}'_3}$.

Now it is easy to see that the composition $F'_2 \circ (F'_1)^{-1}$ is a bijection from $\mathcal{M}'_1$ to $\mathcal{M}'_3$ that satisfies (2.2) and $(F'_2 \circ (F'_1)^{-1})([a_1]_{\mathcal{M}'_1}) = [a_3]_{\mathcal{M}'_3}$, which means that the games $\mathcal{G}_1$ and $\mathcal{G}_3$ are emulationally equivalent. This completes the proof. $\square$

Note: When considered with normal play (as opposed to misère play), the emulational equivalence is a strictly stronger notion than the usual Grundy-equivalence. It is an easy exercise to show that any two games that are emulationally equivalent are also Grundy-equivalent. The converse is not true, as can be seen by the following example. We consider two instances of Nim: first, played on one heap of size 0 (the trivial game, with one position and no moves), and second, played on two heaps, both of size 1. These two games are Grundy-equivalent (their nimbers are 0), but they are not emulationally equivalent.

2.2. **Hackenforb.** We shall now describe a class of games, where each game in the class is determined by a given graph G and a set $\mathcal{F}$ of forbidden connected graphs. For the games from that class we shall use the name *Hackenforb*. The rules are as follows. Players take turns removing an edge of G; if, during this process, a connected component appears that is isomorphic to a graph from $\mathcal{F}$, then that component is erased. The game ends when a player is unable to make a move (that is, there are no more edges left). (Under "normal play condition," the player unable to move loses, while under "misère play condition," that player wins. However, for the purpose of our analysis, the question of the winner is irrelevant.) The Hackenforb game determined by parameters G and $\mathcal{F}$ will be denoted by $\Gamma(G, \mathcal{F})$.

Let us now give a formal definition of Hackenforb.

Definition 2.7. *Given a graph G and a set of connected graphs $\mathcal{F}$, a game of* Hackenforb *determined by G and $\mathcal{F}$, denoted by $\Gamma(G, \mathcal{F})$, is the triple (P, f, G), where:*

- *P is the set of all subgraphs of G that can be obtained by successively removing an edge of G and all the connected components from $\mathcal{F}$ that appear in the remaining graph;*
- *for $G_1, G_2 \in P$, we have that $G_2 \in f(G_1)$ iff G_2 can be obtained by removing an edge from G_1, and additionally removing all the connected components that belong to $\mathcal{F}$.*

Note that we can also assume, without loss of generality, that isolated vertices are deleted as soon as they appear. This convention will be assumed in the rest of the paper. In that case, the game ends when the empty graph is reached (that is, the graph with no vertices); we shall denote it by K_0.

Example. *Let G and $\mathcal{F} = \{G_1, G_2, G_3, G_4\}$ be as in Figure 4. Then, in $\Gamma(G, \mathcal{F})$ we have $P = \{G, G_5, G_6, K_0\}$ (where G_5 and G_6 are also shown in Figure 4). The function f is given by $f(G) = \{K_0, G_5, G_6\}$, $f(G_5) = f(G_6) = \{K_0\}$ and $f(K_0) = \emptyset$.*

3. Emulational equivalence of Hackenforb and various games

In this part we shall show that some well known games are emulationally equivalent to an instance of Hackenforb.

3.1. **Nim.**

The first game we consider is Nim. There are n heaps of coins. Players alternately choose a heap and remove any positive number of coins from that heap. The game ends when there are no coins left.

Let us first represent Nim in the sense of Definition 2.1.

Definition 3.1. *If positive integers $k_1, k_2, \ldots, k_n$ are given, then the game Nim is described by the triple $(P, f, \{k_1, \ldots, k_n\})$, where:*

- *$\{k_1, \ldots, k_n\}$ is a multiset (its elements can be repeated);*
- *P is the set of all multisets $\{s_1, \ldots, s_n\}$ of nonnegative integers for which there exists a permutation σ of the set $\{1, \ldots, n\}$ such that $s_{\sigma(i)} \leqslant k_i$ for all i;*
- *$\{r_1, \ldots, r_n\} \in f_1(\{s_1, \ldots, s_n\})$ iff there exists a permutation θ of the set $\{1, \ldots, n\}$ such that, for some j, $r_{\theta(j)} < s_j$ and $r_{\theta(i)} = s_i$ whenever $i \neq j$.*

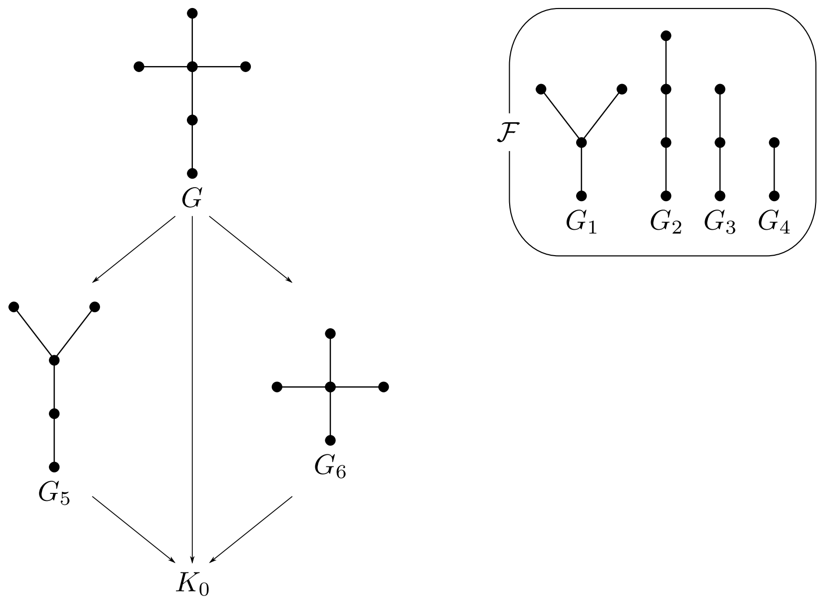

FIGURE 4. An example of Hackenforb.

To emulate Nim played on heaps of sizes $k_1, k_2, \ldots, k_n$, we shall use the instance of Hackenforb $\Gamma(G, \mathcal{F})$ defined as follows. The graph G consists of n connected components, where the i^{th} component is a path of length k_i with two pendant edges attached to one end. The set $\mathcal{F}$ consists of all paths of length less than or equal to $\max\{k_1, \ldots, k_n\} + 1$.

Example. *For Nim with two heaps of size* 3 *and* 5 *the corresponding graph will be* G *from Figure 5, and forbidden graphs will be all paths of length less than or equal to* 6.

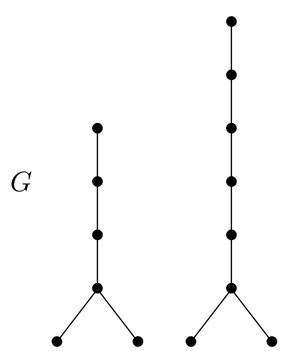

FIGURE 5. The emulating graph for Nim with heaps of size 3 and 5.

The idea of the emulation is the following one. Each connected component of G represents one heap. Removing any edge from any component will leave either a path (which is forbidden and thus the whole component will be erased), or again a path with two pendant edges attached to one end (plus eventually another path that will be erased); in the latter case, the number of edges in the remaining graph, reduced by 2 (that is, not counting the two attached edges), will stand for the remaining number of coins on the corresponding heap.

Example. *If there are two heaps of size* 3 *and* 5*, and a player takes* 2 *coins from the heap of size* 5*, in the corresponding Hackenforb game this is represented by a move shown in Figure 6 left; namely, the red edge is the edge that is being removed, and after that the green part will be erased because it is forbidden.*

Figure 6 right shows a move in the Hackenforb game that represents taking the whole heap of size 3 *in Nim.*

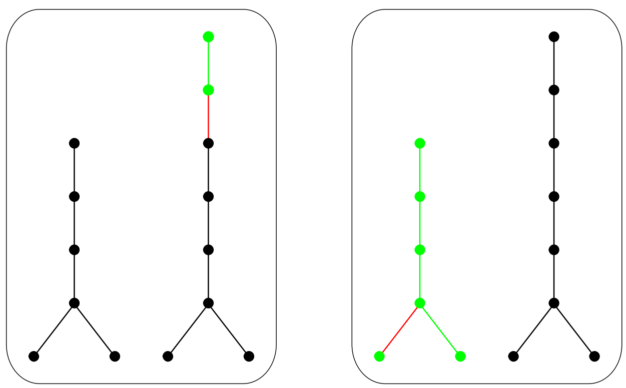

FIGURE 6. Two moves in Hackenforb that emulate removing 2 coins from the heap of size 5 (left) and removing a whole heap of size 3 (right).

We now give a formal proof of the emulational equivalence. Let $\Gamma(G, \mathcal{F}) = (P^*, f^*, G)$. To any graph a from P^*, we assign the multiset $\{a_1, a_2, \dots, a_n\}$ that represents the number of edges in each connected component of a decreased by 2 (if there are less than n connected components, then we include a number of zeros in the multiset, such that the total number of elements is n); note that all the connected components of a have at least three edges, and thus all the elements from the multiset are nonnegative integers.

Theorem 3.2. *The game Nim is emulationally equivalent to the described instance of Hackenforb.*

Proof. Let $\mathcal{P}$ and $\mathcal{P}^*$ be the trivial partitions of the games Nim and $\Gamma(G, \mathcal{F})$, respectively. Let $F : \mathcal{P} \to \mathcal{P}^*$ be defined by:

$$F(\{s_1, \dots, s_n\}) = \{a\},$$

where a is the graph from P^* whose corresponding multiset is $\{s_1, \dots, s_n\}$. The defined function is clearly a bijection. The corresponding multiset for G is $\{k_1, \dots, k_n\}$, so $F(\{k_1, \dots, k_n\}) = \{G\}$. Let us prove that it satisfies the condition (2.2).

($\Rightarrow$): Let $\{s_1, \dots, s_n\}, \{r_1, \dots, r_n\} \in \mathcal{P}$ and $\{r_1, \dots, r_n\} \in f(\{s_1, \dots, s_n\})$. We prove that $b \in f^*(a)$, where $F(\{s_1, \dots, s_n\}) = \{a\}$ and $F(\{r_1, \dots, r_n\}) = \{b\}$. Without loss of generality, we can assume that $r_1 < s_1$ and $r_i = s_i$ whenever $2 \leqslant i \leqslant n$. Then, considering the connected component of a that has $s_1 + 2$ edges, by one move we can remove $s_1 - r_1$ edges from it and thus get the position b.

($\Leftarrow$): Suppose now that for two graphs $a, b \in P^*$ we have $b \in f^*(a)$. Let $F(\{s_1, \dots, s_n\}) = \{a\}$ and $F(\{r_1, \dots, r_n\}) = \{b\}$. We need to prove $\{r_1, \dots, r_n\} \in f(\{s_1, \dots, s_n\})$. By one move in Hackenforb, we decrease the number of edges in one component or delete the whole component. That means that the observed multisets differ in one element; without loss of generality, let that be the first one, in particular, $r_1 < s_1$. Now in the game Nim we can remove $s_1 - r_1$ coins (possibly the whole heap) from the heap of size s_1 and thus get a position $\{r_1, \dots, r_n\}$. □

3.2. **Subtraction game.** The next game we shall deal with is a game similar to Nim, usually called the Subtraction game. The Subtraction game is defined by two positive integers n and k, where $k \leqslant n$. The game is played between two players and a heap of n coins. Two players alternate taking any number from 1 to k coins from the heap. The game ends when there are no coins left. (Note: this is the most basic version of the subtraction game. Some generalized versions also exist in the literature, but we do not consider them here.)

Now we want to express the Subtraction game in the sense of Definition 2.1.

Definition 3.3. *The Subtraction game with parameters $n, k \in \mathbb{N}$, $k \leqslant n$, is the triple (P, f, n), where $P = \{0, 1, \dots, n\}$ and $f : P \to \mathcal{P}(P)$ is defined by*

$$b \in f(a) \Leftrightarrow 1 \leqslant a - b \leqslant k.$$

Let n and k be given, and let $N = n - k + 3$. Let the graph G be obtained by taking N paths of length k and identifying one end of each of them with a common vertex (we shall call it the "central vertex"). Let the set $\mathcal{F}$ consist of all paths of length less than k, as well as all the subgraphs of G with less than or equal to $kN - n$ edges. We shall prove that the Subtraction game with parameters n and k is emulationally equivalent to $\Gamma(G, \mathcal{F})$.

Example. *For $n = 6$ and $k = 4$ the corresponding graph G is given in Figure 7 and forbidden graphs are all paths of length less than 4 and all subgraphs of G with no more than 14 edges.*

The Subtraction game will be emulated by Hackenforb in the following way: removing an edge from G after which G remains with s edges less

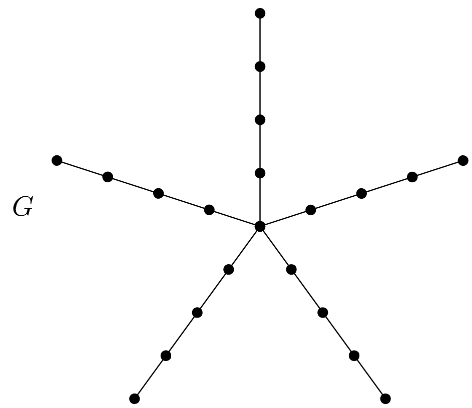

FIGURE 7. The emulating graph for Subtraction game with $n = 6$ and $k = 4$.

(the edge that has been removed, plus the edges forming the forbidden part after that move) represents removal of s coins from the heap in Subtraction game. Before we prove this formally, we describe a congruential partition of $\Gamma(G, \mathcal{F})$ that we are going to use.

Proposition 3.4. *Let* $\Gamma(G, \mathcal{F}) = (P^*, f^*, G)$. *Let*

$$\mathcal{P}^* = \{S_p : p \in \{0, 1, \ldots, n-1\}\} \cup \{\{K_0\}\},$$

where S_p *is the set of all positions from* P^* *in which the number of edges is exactly* $kN - p$. *Then* $\mathcal{P}^*$ *is a congruential partition.*

Proof. It is clear that $\mathcal{P}^*$ is a partition of the set P^*. Therefore, we need to prove that the condition (2.1) is true.

Assume first $a \in S_p$, $b \in S_r$ and $b \in f^*(a)$, $p, r \in \{0, 1, \ldots, n-1\}$ (that is, $b \neq K_0$ in this case). Let $c \in S_p$. We need to prove that $f^*(c)$ contains a member of S_r. There are $kN - p$ edges in a and c, and $kN - r$ edges in b. Note that a move from a to b removes a total of $kN - p - (kN - r)$, that is, $r - p$ edges (where one of them is the edge that has been erased, and the rest of them form a forbidden path). It is enough to prove that, in the graph c, there exists a path of length at least $r - p$ starting from the central vertex; indeed, in that case, we shall be able to remove a total of $r - p$ edges from that path, which leads to a position from S_r. Since in the initial position there were N paths starting from the central vertex, each of length k, and a total of p edges have been removed, we conclude that there exists a path of length at least $k - \lfloor \frac{p}{N} \rfloor$, which is at least $k - \frac{p}{N}$. If $p < N$, then there exists a path of length k, and the conclusion in that case is trivial. Assume now $p \geqslant N$. We want to prove $k - \frac{p}{N} \geqslant r - p$. And indeed:

$$\begin{aligned}\left(k - \frac{p}{N}\right) - (r - p) &= k - r + p\left(1 - \frac{1}{N}\right)\\ &\geqslant k - r + N\left(1 - \frac{1}{N}\right)\\ &= k - r + N - 1\\ &= k - r + (n - k + 3) - 1\\ &= n - r + 2\\ &> 0.\end{aligned}$$

Assume now that for some p, $p \in \{0, 1, \ldots, n-1\}$, and $a \in S_p$ we have $K_0 \in f^*(a)$. Then there is an edge in a whose removal results in a path of length less than k, and a graph with a total of no more than $kN - n$ edges (both of which are forbidden components). This implies that a has a total of no more than $kN - n + k$ edges, that is, $p \geqslant n - k$. It is enough to prove that, for any $c \in S_p$, there exists a path in c of length at least $n - p$ (which is no more than k) starting from the central vertex. As in the previous paragraph, this reduces to $k - \frac{p}{N} \geqslant n - p$, which we show in an analogous way. □

Theorem 3.5. *The Subtraction game is emulationally equivalent to the described instance of Hackenforb.*

Proof. Let $\mathcal{P}$ be the trivial partition of the Subtraction game, and $\mathcal{P}^*$ the partition of $\Gamma(G, \mathcal{F})$ from the previous proposition. Define $F : \mathcal{P} \to \mathcal{P}^*$ by

$$F(\{a\}) = S_{n-a}$$

for $a \in \{1, \ldots, n\}$ and $F(\{0\}) = \{K_0\}$. The function F is clearly a bijection and by its definition we have $F(\{n\}) = S_0$, while $S_0 = \{G\}$ (note that G has exactly kN edges). We are left to prove the condition (2.2).

($\Rightarrow$): Let $\{a\}, \{b\} \in \mathcal{P}$ and $b \in f(a)$. We should prove $S_{n-b} \in f^*(S_{n-a})$ if $b > 0$, respectively $\{K_0\} \in f^*(S_{n-a})$ if $b = 0$. Since $b \in f(a)$, we have $1 \leqslant a - b \leqslant k$.

First, let $b > 0$. Let c be an arbitrary graph from S_{n-a}. Then we have to show that it is possible to remove exactly $a - b$ edges from c, which can be proved in exactly the same manner as in the previous proposition.

If $b = 0$, that is, $0 \in f(a)$, then $1 \leqslant a \leqslant k$, and we can again choose any graph from S_{n-a} and as in the previous proposition show that we can reach K_0 in one move.

($\Leftarrow$): Let now $S_{n-b} \in f^*(S_{n-a})$ for some $a, b \in \{1, \ldots, n\}$; in particular, $b \neq 0$. Any graph from S_{n-a} has a total of $kN - (n - a)$ edges, while any graph from S_{n-b} has a total of $kN - (n - b)$ edges. By the definition of f^*, in one move we can remove at least one and at most a total of k edges (a chosen edge, plus the forbidden path). Therefore, $1 \leqslant kN - (n-a) - (kN - (n-b)) \leqslant k$, that is, $1 \leqslant a - b \leqslant k$, which gives $b \in f(a)$.

Finally, let $\{K_0\} \in f^*(S_{n-a})$ for some a, $a \in \{1, \dots, n\}$. We shall prove that any graph c from P^* such that $K_0 \in f^*(c)$ has at most $kN - n + k$ edges. For that, we shall first prove that after any move on c, there may appear at most one forbidden component that is a path of length less than k. Indeed, since $N \geqslant 3$, one of the remaining components will always have a vertex of degree greater than 2, with the only exception of the case when there are only three paths left and we choose to remove one of the edges incident to the central vertex; however, if in this case c splits into two paths both of length less than k, that means that c has less than $2k$ edges, that is, more than $kN - 2k$ edges have been removed from the beginning of the game, but since the inequality $kN - 2k \geqslant n$ is equivalent to $(n-k)(k-1) \geqslant 0$, which is true, we conclude that c is a forbidden graph, which is a contradiction. This further means that the move that leads from c to K_0 leaves a forbidden path of length less than k, and a forbidden component with no more than $kN - n$ edges; altogether, c has at most $kN - n + k$ edges. Therefore, $a \leqslant k$, which implies $0 \in f(a)$. $\square$

3.3. **Notakto.** In this subsection we shall prove that an impartial variant of tic-tac-toe called Notakto is emulationally equivalent to an instance of Hackenforb. Notakto is played on one or more 3×3 boards (the same board as for tic-tac-toe), and on each turn, a player chooses a board (if there are more than one of them) and puts an X in any empty space on it. When three X's in a row appear on the same board, that board cannot be used for playing anymore. The game ends when there are no more available moves (and the game is usually played under misère play convention, that is, the player who made the last move loses the game).

We shall first show that a single-board Notakto is emulationally equivalent to an instance of Hackenforb, and after that we shall generalize the theorem for multiple-boards Notakto.

Note that no more than 6 moves can be played on any 3×3 board without ending the game (if 7 X's are written, there will be three in a row among them); see Figure 8. We shall enumerate the fields of a 3×3 board as in

×	×	
×		×
	×	×

FIGURE 8. A placement of 6 X's with no three in a row.

Figure 9. For a partially filled board T and $n \in \{1, \dots, 9\}$, we shall write

1	2	3
4	5	6
7	8	9

FIGURE 9. Enumeration of the fields on the board.

$T(n) = X$ iff X is written in the field n.

Definition 3.6. *For two* 3×3 *boards* T_1 *and* T_2 *we say that* T_1 *is a* subboard *of* T_2*, denoted by* $T_1 \leqslant T_2$*, if and only if*

$$(\forall n \in \{1, \dots, 9\})(T_1(n) = X \Rightarrow T_2(n) = X).$$

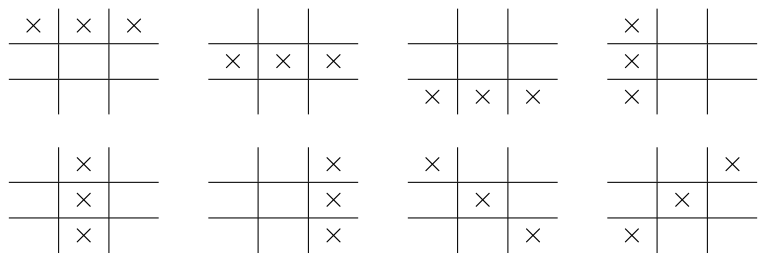

FIGURE 10. The set E.

Let E be the set of all the boards from Figure 10. We shall now express Notakto in the sense of Definition 2.1.

Definition 3.7. *The game Notakto (on one board) is described by the triple* (P, f, T_0)*, where:*

- P *is the set of all* 3×3 *boards partially filled by* X*'s, which either do not contain any subboard from* E*, or an* X *can be removed in such a way that the resulting board does not contain any subboard from* E*;*
- $f : P \to \mathcal{P}(P)$ *is defined by*
$$T_2 \in f(T_1) \iff (\forall T' \in E)\neg(T' \leqslant T_1),$$
and $T_1 \leqslant T_2$*, and* T_2 *has one more* X *than* T_1 *has;*
- T_0 *is the empty board.*

We shall now describe a congruential partition of Notakto that we are going to use. In short, all the ending positions will be together in one class, and all the other positions will be alone in their class.

Proposition 3.8. *Let*

$$\begin{aligned}\mathcal{P} =&\{\{T\} : T \in P \wedge (\forall T' \in E)\neg(T' \leqslant T)\}\\ &\cup \{\{T : T \in P \wedge (\exists T' \in E)(T' \leqslant T)\}\}.\end{aligned}$$

Then $\mathcal{P}$ *is a congruential partition of Notakto.*

Proof. $\mathcal{P}$ is obviously a partition of P. The condition (2.1) is clearly fulfilled whenever A is a singleton, and it is trivially true whenever A is the class consisting of all ending positions. □

Let us now describe the instance of Hackenforb that is emulationally equivalent to Notakto. Let G be the graph obtained by taking the complete graphs $K_8, K_9, \dots, K_{16}$ and identifying one vertex of each of them with a common vertex, called the "central vertex." Note that, since the game lasts at most 7 moves, the graph obtained after each move will always be connected (with the possible exception of obtaining one isolated vertex after the last move). Also, it will always be possible to identify the central vertex (that shall be the vertex of the greatest degree). Finally, for each edge of any of the graphs obtained during the game, it will always be possible to tell to which of the originally joined complete graphs it belongs (indeed, removing the central vertex leaves 9 connected components, which correspond to the 9 original complete graphs in increasing size; note that $|E(K_{i+1})| - |E(K_i)| \geqslant 8$ whenever $i \geqslant 8$, and therefore the order remains invariant during the game). Therefore, for each graph obtained during the game, we can determine how many edges have been deleted from each of those complete graphs. That way, if G' is any graph that is possible to obtain during the game, we can uniquely assign a nonuple $(x_1, \dots, x_9)$ from $\{0, \dots, 7\}^9$ to G' such that x_i is the number of removed edges from the subgraph K_{i+7}. For each such nonuple we have $\sum_{i=1}^{9} x_i \leqslant 7$.

Definition 3.9. *Let $(x_1, \dots, x_9) \in \{0, \dots, 7\}^9$ with $\sum_{i=1}^{9} x_i \leqslant 7$. If $x_i > 1$ for some i, let k be the smallest positive integer such that $x_{i+k} = 0$ (where indices are taken cyclically modulo 9). Consider the transformation that maps the observed nonuple to $(x'_1, \dots, x'_9)$, where $x'_i = x_i - 1$, $x'_{i+k} = 1$, and $x'_j = x_j$ otherwise. The nonuple obtained by iterating this transformation as long as possible (that is, as long as there are elements greater than 1) is called the* standard nonuple *for $(x_1, \dots, x_9)$, and is denoted by $(x_1^0, \dots, x_9^0)$. (Note that we have $\sum_{i=1}^{9} x_i^0 = \sum_{i=1}^{9} x_i$.)*

The following lemma shows that the standard nonuple is well-defined, that is, that the choices that we make for i in each step do not affect the final result. This will be proved by giving an equivalent definition of the standard nonuple that is not choice-dependent.

Lemma 3.10. *Let a nonuple $(x_1, \dots, x_9) \in \{0, \dots, 7\}^9$ with $\sum_{i=1}^{9} x_i \leqslant 7$ be given. Then its standard nonuple $(x_1^0, \dots, x_9^0)$ is given by*

$$x_i^0 = 1 \Leftrightarrow (\exists k \in \{0, 1, \dots, 6\}) \left(\sum_{t=0}^{k} x_{i-t} > k \right) \tag{3.1}$$

for $i \in \{1, \dots, 9\}$ (where indices are taken cyclically modulo 9).

Proof. Let $(x_1, \dots, x_9) \in \{0, \dots, 7\}^9$ with $\sum_{i=1}^{9} x_i \leqslant 7$. The proof is easy if each $x_i \in \{0, 1\}$ (then $x_i^0 = x_i$ under both definitions).

Therefore, let $x_j > 1$ for some j, and let r be the smallest positive integer such that $x_{j+r} = 0$ (here and onward, we assume that indices are taken cyclically modulo 9). We note that $r \leqslant 6$, since $\sum_{i=1}^{9} x_i \leqslant 7$. Let a nonuple $(y_1, \dots, y_9)$ be such that $y_j = x_j - 1$, $y_{j+r} = 1$ and $y_i = x_i$ otherwise. It is enough to prove that $(y_1^0, \dots, y_9^0) = (x_1^0, \dots, x_9^0)$, where these nonuples are defined as in (3.1). Fix i, $i \in \{1, \dots, 9\}$, and let us show that $x_i^0 = y_i^0$.

Assume first $x_i^0 = 1$. Then there exists k, $k \in \{0, \dots, 6\}$, such that $\sum_{t=0}^{k} x_{i-t} > k$. The sum $\sum_{t=0}^{k} y_{i-t}$ can be:

- equal to $\sum_{t=0}^{k} x_{i-t}$ (if neither y_j nor y_{j+r} appear in the sum, or if both of them appear); in this case we have $y_i^0 = x_i^0 = 1$;
- equal to $\sum_{t=0}^{k} x_{i-t} + 1$ (if y_{j+r} appears in the sum and y_j does not); in this case, $\sum_{t=0}^{k} y_{i-t} > k+1 > k$, and therefore $y_i^0 = 1$;
- equal to $\sum_{t=0}^{k} x_{i-t} - 1$ (if y_j appears in the sum and y_{j+r} does not); in this case, i is an index between j and $j+r-1$ (in cyclic order), but then $x_i \geqslant 1$ and $y_i = x_i \geqslant 1$ if $i \neq j$ (by the choice of r), respectively $y_i = x_i - 1 \geqslant 1$ if $i = j$ (since $x_j > 1$), which in both cases implies $y_i^0 = 1$.

Assume now $y_i^0 = 1$. Let k, $k \in \{0, \dots, 6\}$, be such that $\sum_{t=0}^{k} y_{i-t} > k$. If y_j appears among these summands, or y_{j+r} does not appear, then $\sum_{t=0}^{k} x_{i-t} \geqslant \sum_{t=0}^{k} y_{i-t} > k$, and thus $x_i^0 = 1$. Assume that y_{j+r} appears among the summands and y_j does not. This means that the indices $j, i-k, j+r, i$ are exactly in this cyclic order (and the first two must be different). By the definitions of j and r, all the numbers $y_j, y_{j+1}, \dots, y_{i-k-1}$ are positive. Therefore,

$$\sum_{t=0}^{i-j} y_{i-t} \geqslant i-k-j+\sum_{t=0}^{k} y_{i-t} > i-j.$$

Since the left-hand side is bounded from above by 7, we conclude $i-j \leqslant 6$. But then, since

$$\sum_{t=0}^{i-j} x_{i-t} = \sum_{t=0}^{i-j} y_{i-t} > i-j,$$

we conclude $x_i^0 = 1$, which was to be proved. □

Example. *Starting from the nonuple* $(0,0,3,0,1,0,0,0,2)$*, we have:*

$$\begin{aligned}(0,0,3,0,1,0,0,0,2) &\mapsto (0,0,2,1,1,0,0,0,2)\\ &\mapsto (0,0,1,1,1,1,0,0,2)\\ &\mapsto (1,0,1,1,1,1,0,0,1).\end{aligned}$$

Therefore, $(1,0,1,1,1,1,0,0,1)$ *is the standard nonuple for the observed nonuple.*

Now we are ready to define the set of forbidden graphs. Let

$$\begin{aligned}E' = \{&(1,1,1,0,0,0,0,0,0),(0,0,0,1,1,1,0,0,0),(0,0,0,0,0,0,1,1,1),\\ &(1,0,0,1,0,0,1,0,0),(0,1,0,0,1,0,0,1,0),(0,0,1,0,0,1,0,0,1),\\ &(1,0,0,0,1,0,0,0,1),(0,0,1,0,1,0,1,0,0)\}.\end{aligned}$$

For two nonuples $(x_1, \dots, x_9), (y_1, \dots, y_9) \in \{0,1\}^9$, we shall say that the second one *contains* the first one if and only if $x_i \leqslant y_i$ for all i. Recall that each subgraph of G with at least $|E(G)| - 7$ edges has its assigned nonuple, and each nonuple has the corresponding standard nonuple. We define the set of forbidden graphs $\mathcal{F}$ to consist of all subgraphs G' of G such that $|E(G)| - |E(G')| \leqslant 7$ and the corresponding standard nonuple for G' contains at least one nonuple from E'.

We have thus defined our instance of Hackenforb.

Proposition 3.11. *Let $\Gamma(G, \mathcal{F}) = (P^*, f^*, G)$. Let S be the set of all standard nonuples from $\{0,1\}^9$ that do not contain any member of E'. Let*

$$\mathcal{P}^* = \{S_{x_1,\dots,x_9} : (x_1, \dots, x_9) \in S\} \cup \{\{K_0\}\},$$

where $S_{x_1,\dots,x_9}$ contains all the positions from P^* whose corresponding standard nonuple is $(x_1, \dots, x_9)$. Then $\mathcal{P}^*$ is a congruential partition.

Proof. Clearly, $\mathcal{P}^*$ is a partition of P^*. Let us show the condition (2.1). Let $a = (x_1, \dots, x_9) \in S_{x_1^0,\dots,x_9^0}$ and $b = (y_1, \dots, y_9) \in S_{y_1^0,\dots,y_9^0}$, and let $b \in f^*(a)$. Then b differs from a at exactly one coordinate, which is decreased by 1. Let us show that their corresponding standard nonuples also differ by one at one coordinate. Since $x_i^0 = 1$ means that there exists k, $k \in \{0, \dots, 6\}$, such that $\sum_{t=0}^{k} x_{i-t} > k$, and the corresponding sum of y_i's is no less, then follows $y_i^0 = 1$. The claim now follows from $\sum_{i=1}^{9} y_i^0 = \sum_{i=1}^{9} y_i = \sum_{i=1}^{9} x_i + 1 = \sum_{i=1}^{9} x_i^0 + 1$.

Let i be the coordinate for which $y_i^0 = 1$ and $x_i^0 = 0$. The definition of a standard nonuple gives that, since $x_i^0 = 0$, then each nonuple from $S_{x_1^0,\dots,x_9^0}$ also has 0 at the i^{th} coordinate. Therefore, for each c, $c \in S_{x_1^0,\dots,x_9^0}$, there are no edges removed from the block K_{7+i}. But then, removing an edge from that block we reach a position whose corresponding standard nonuple is $(y_1^0, \dots, y_9^0)$, as needed.

The same proof holds if $b = K_0$ instead of $b \in S_{y_1^0,\dots,y_9^0}$ (the only difference is that the corresponding standard nonuple obtained after the move from a now contains a nonuple from E', and the above argument is the same). And finally, if $a = K_0$, the conclusion is trivial. This completes the proof. □

The idea of emulation is the following one: removing an edge from K_{7+i} in G emulates putting an X in the field i or, if that field is occupied, in the first available field after it in the cyclic order.

Theorem 3.12. *Notakto is emulationally equivalent to an instance of Hackenforb.*

Proof. We define a bijection F between the congruential partitions $\mathcal{P}$ and $\mathcal{P}^*$ as follows. If $T \in P$ is such that $(\forall T' \in E)\neg(T' \leqslant T)$, then $\{T\} \in \mathcal{P}$ and we define

$$F(\{T\}) = S_{x_1,\dots,x_9}, \qquad \begin{aligned} &\text{where } x_i = 1 \text{ if } T(i) = X \\ &\text{and } x_i = 0 \text{ if } T(i) \text{ is empty;} \end{aligned}$$

and otherwise

$$F(\{T : T \in P \wedge (\exists T' \in E)(T' \leqslant T)\}) = \{K_0\}.$$

It is easy to see that F is a bijection between $\mathcal{P}$ and $\mathcal{P}^*$. By the definition of F, $F(\{T_0\}) = S_{0,\dots,0} = \{G\}$. We are left to prove the condition (2.2).

($\Rightarrow$): Let first $\{T_1\}, \{T_2\} \in \mathcal{P}$ and $T_2 \in f(T_1)$. Let $F(\{T_1\}) = S_{x_1,\dots,x_9}$ and $F(\{T_2\}) = S_{y_1,\dots,y_9}$. We need to prove $S_{y_1,\dots,y_9} \in f^*(S_{x_1,\dots,x_9})$, for which it is enough to find $a \in S_{x_1,\dots,x_9}$ and $b \in S_{y_1,\dots,y_9}$ such that $b \in f^*(a)$. Since $T_2 \in f(T_1)$, it follows that $(x_1,\dots,x_9)$ and $(y_1,\dots,y_9)$ differ at exactly one coordinate, say i_0, that is, $x_{i_0} = 0$ and $y_{i_0} = 1$. Let a be a subgraph of G with one removed edge from K_{7+i} whenever $x_i = 1$, and let b a subgraph of a with one edge further removed from K_{7+i_0}. Then clearly $b \in f^*(a)$, as needed.

We are left to check the case when a position T_1 is given, $\{T_1\} \in \mathcal{P}$, such that there is a move from T_1 leading to an ending position. Let T_2 be that ending position, and let it be obtained from T_1 by writing an X in the field i_0. Let $F(\{T_1\}) = S_{x_1,\dots,x_9}$ (where $x_{i_0} = 0$). We need to prove that there exists $a \in S_{x_1,\dots,x_9}$ such that there is a move from a to the empty graph. Let $(y_1,\dots,y_9)$ be the corresponding standard nonuple for T_2. Since T_2 contains a subboard from E, the nonuple $(y_1,\dots,y_9)$ must contain a nonuple from E'. Let a be a subgraph of G with one removed edge from K_{7+i} whenever $x_i = 1$. Note that removing an edge from K_{7+i_0} in a leads to a graph whose corresponding standard nonuple is $(y_1,\dots,y_9)$, but since $(y_1,\dots,y_9)$ contains a nonuple from E', the obtained graph is forbidden; in other words, $K_0 \in f^*(a)$, which was to be proved.

($\Leftarrow$): Let now $S_{y_1,\dots,y_9} \in f^*(S_{x_1,\dots,x_9})$ for some nonuples $(x_1,\dots,x_9)$, $(y_1,\dots,y_9)$ from S, and let $F(\{T_1\}) = S_{x_1,\dots,x_9}$, $F(\{T_2\}) = S_{y_1,\dots,y_9}$. We need to prove $\{T_2\} \in f(\{T_1\})$, that is, $T_2 \in f(T_1)$. But it is clear that the nonuples $(x_1,\dots,x_9)$ and $(y_1,\dots,y_9)$ differ at precisely one coordinate, and writing an X in the corresponding field is a move from T_1 to T_2, which was to be proved.

We are left to check the case $\{K_0\} \in f^*(S_{x_1,\dots,x_9})$ for a nonuple $(x_1,\dots,x_9)$ from S. Let $F(\{T_1\}) = S_{x_1,\dots,x_9}$. We need to prove that there exists T_2 such that $T_2 \in f(T_1)$ and T_2 contains a subboard from E. Note that, because of $\{K_0\} \in f^*(S_{x_1,\dots,x_9})$, there exists a coordinate i_0 in $(x_1,\dots,x_9)$ such that, by changing x_{i_0} from 0 to 1, we reach a nonuple that contains a nonuple from E'. But then putting an X in the field i_0 is a move that leads from T_1 to a position that contains a subboard from E, which completes the proof. □

Finally, we show that emulational equivalence of multiple-boards Notakto to an instance of Hackenforb easily follows from the single-board case.

Theorem 3.13. *Multiple-boards Notakto is emulationally equivalent to an instance of Hackenforb.*

Proof. Suppose we have n boards. We consider the instance of Hackenforb $\Gamma(G', \mathcal{F})$, where the graph G' has n connected components each of which is isomorphic to G (from the single-board case); and the set of forbidden graphs is the same $\mathcal{F}$ from the single-board case. The emulational equivalence now follows from the previous theorem. □

3.4. **Chomp.** Chomp is a combinatorial game played on a rectangular board whose upper right field is "poisoned." On each move, a player chooses a non-eaten field and eats all the fields that are below and to the left of the chosen one (including it). The loser is the player who eats the poisoned field. In this subsection we show that Chomp is emulationally equivalent to an instance of Hackenforb.

Let us first express Chomp in the sense of Definition 2.1. Assume that there are n rows and k columns. Any position in the game will be represented by a k-tuple $(a_1, \dots, a_k)$, $0 \leqslant a_i \leqslant n$, where a_i stands for the number of eaten fields in the i^{th} column (where columns are enumerated by $1, \dots, k$ from left to right). Note that the rules of Chomp imply that the elements $a_1, \dots, a_k$ will always be in nonincreasing order (and vice versa: each k-tuple $(a_1, \dots, a_k)$ whose elements are in nonincreasing order represents a valid position in the game). In this language, a move in the game consists of replacing a few consecutive elements in the tuple by a common value, greater than all of them, with an additional requirement that the obtained tuple still has to be in nonincreasing order.

Definition 3.14. *If positive integers k and n are given, then the game Chomp is described by the triple $(P, f, (0, \dots, 0))$, where:*

- *P is the set of all nonincreasing k-tuples from $\{0, 1, \dots, n\}^k$;*
- *for $(a_1^1, \dots, a_k^1), (a_1^2, \dots, a_k^2) \in P$, we have*

$$(a_1^2, \dots, a_k^2) \in f((a_1^1, \dots, a_k^1))$$

if and only if there exist $u \in \{1, \dots, k\}$, $v \in \{0, \dots, k-1\}$ and $t \in \{1, \dots, n\}$ such that

$$(\forall i)((i < u \vee u + v < i \Rightarrow a_i^1 = a_i^2) \wedge (u \leqslant i \leqslant u + v \Rightarrow a_i^1 < a_i^2 = t)).$$

Example. *In Figure 11 we show some positions in the game Chomp, their corresponding tuples, and the possible moves between them.*

From now on, we assume that positive integers k and n are fixed. In order to prove that Chomp is emulationally equivalent to an instance of Hackenforb, we shall first introduce an auxiliary game, called *Auxie,* and then prove that Chomp is emulationally equivalent to Auxie, as well as that Auxie is emulationally equivalent to an instance of Hackenforb.

3.4.1. *Chomp and Auxie.* We here define the game Auxie and show that it is emulationally equivalent to Chomp. In the definition below we, after defining what is necessary in a formal manner, explain the same in an informal manner, which is probably more intuitive to the reader. These explanations are marked by the symbol "$\triangleleft$". They will be referring to a board with n rows and k columns, where rows are numbered from 1 to n from bottom to top, and columns from 1 to k from left to right, and in each field there is written either 0 or 1.

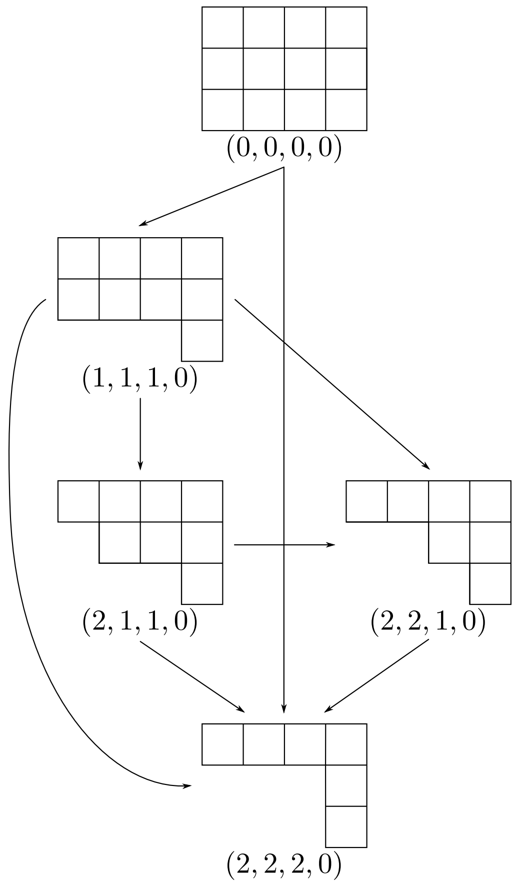

FIGURE 11. Some positions and moves in Chomp with 4 columns and 3 rows.

Definition 3.15. *The game Auxie is represented by the triple* (P', f', b_0), *where:*

- *the set* P' *consists of* k*-tuples whose elements are* n*-tuples of* 0*s and* 1*s (in other words:* $P' \subseteq \{\{0,1\}^n\}^k$*) such that*

$$((x_1^1, \dots, x_n^1), \dots, (x_1^k, \dots, x_n^k)) \in P'$$

if and only if

$$(\forall l)(\forall i)(\forall j)((j \geqslant l \Rightarrow x_j^i = 0) \Rightarrow (\forall m > i)(j \geqslant l \Rightarrow x_j^m = 0)); \tag{3.2}$$

$\triangleleft$ *In other words: if there is a vertical sequence of* 0*s from some field upwards all the way to the top, then all the fields to the right of this sequence must also contain* 0*s.*

- *for* $((x_1^1, \dots, x_n^1), \dots, (x_1^k, \dots, x_n^k)), ((y_1^1, \dots, y_n^1), \dots, (y_1^k, \dots, y_n^k)) \in P'$, *we have*

$$((y_1^1, \dots, y_n^1), \dots, (y_1^k, \dots, y_n^k)) \in f'(((x_1^1, \dots, x_n^1), \dots, (x_1^k, \dots, x_n^k)))$$

if and only if there exist $u \in \{1, \ldots, k\}$, $v \in \{0, \ldots, k-1\}$, $t \in \{1, \ldots, n\}$ *such that*

$$(\forall i \geqslant u)(\forall j \geqslant t)(x_j^i = 0)$$

and

$$y_j^i = \begin{cases} 1, & \text{if } u \leqslant i \leqslant u+v \text{ and } j = t; \\ x_j^i, & \text{otherwise;} \end{cases} \tag{3.3}$$

◁ *In other words: making a move consists of choosing a horizontal sequence of* 0*s such that everything above and to the right of them are* 0*s, and replacing this sequence of* 0*s by* 1*s. (Note: it is assumed that the position thus obtained must satisfy the condition from the previous bullet point; otherwise, the considered move is impossible. This assumption is encompassed in the requirement that the considered tuples belong to* P'*.)*

- $b_0 = ((0, 0, \ldots, 0), \ldots, (0, 0, \ldots, 0))$.

The idea of emulation is the following one: the n-tuples represent columns (from left to right) of the Chomp table. The number 1 in a certain position means that the corresponding field in Chomp has been removed (in some stage of the game) either by being directly chosen, or because a field in the same row and to the right of it has been chosen (therefore, fields that are removed solely as a consequence of the fact that some field *above* them is removed will *not* be marked by 1). In particular, the move in Auxie defined by (3.3) corresponds to the move in Chomp in which a field in the t^{th} row and the $(u+v)^{\text{th}}$ column is chosen. Note that the pattern of 0s and 1s in Auxie can provide some information about the sequence of moves played in Chomp in order to reach the concerned position, though it does not necessarily determine that sequence uniquely (that is, it is still possible that different sequences of moves in Chomp result in the same pattern in the corresponding game of Auxie).

Example. *In Figure 12 we show some possible positions in the game Auxie and the possible moves between them. The positions are shown in the form of the boards described above (for example, the position shown bottom-right corresponds to the tuple* $((1,1,0),(0,1,0),(1,1,0),(0,0,0))$*). Positions encircled together are those that belong to the same class in the congruential partition of Auxie that we are going to use (and that will be introduced in a moment). All the classes except the one at the bottom are complete as shown (that is, they do not contain any other positions not shown in the figure). Note that dashed lines represent moves between positions in Auxie, while solid lines represent moves between classes. The classes shown here are precisely those corresponding to the positions in Chomp shown in Figure 11 (with respect to the emulational equivalence that will be defined in Theorem 3.17); in particular, the class denoted by* S_{a_1,a_2,a_3,a_4} *corresponds to the position* (a_1, a_2, a_3, a_4) *in Chomp (more precisely, to the class* $\{(a_1, a_2, a_3, a_4)\}$

in the trivial partition of Chomp, since the trivial partition is the one that we are going to use).

Let us now describe a congruential partition of Auxie that we are going to use. Let e_j, for $j \in \{1, \dots, n\}$, denote the n-tuple $(0, \dots, 0, 1, 0, \dots, 0)$ where there is 1 at the j^{th} coordinate, and 0 at all the rest. Let e_0 denote the n-tuple $(0, \dots, 0)$.

Proposition 3.16. a) *Let $j_1, j_2, \dots, j_k$ be given, $0 \leqslant j_i \leqslant n$. Then $(e_{j_1}, e_{j_2}, \dots, e_{j_k}) \in P'$ if and only if $j_1 \geqslant j_2 \geqslant \dots \geqslant j_k$.*

b) *For $n \geqslant j_1 \geqslant j_2 \geqslant \dots \geqslant j_k \geqslant 0$, let $S_{j_1,j_2,\dots,j_k}$ denote the subset of P' defined by:*

$$((y_1^1, \dots, y_n^1), \dots, (y_1^k, \dots, y_n^k)) \in S_{j_1,j_2,\dots,j_k}$$

if and only if

$$(\forall i)\big((\forall p)(p > j_i \Rightarrow y_p^i = 0) \wedge (j_i = 0 \vee y_{j_i}^i = 1)\big)$$

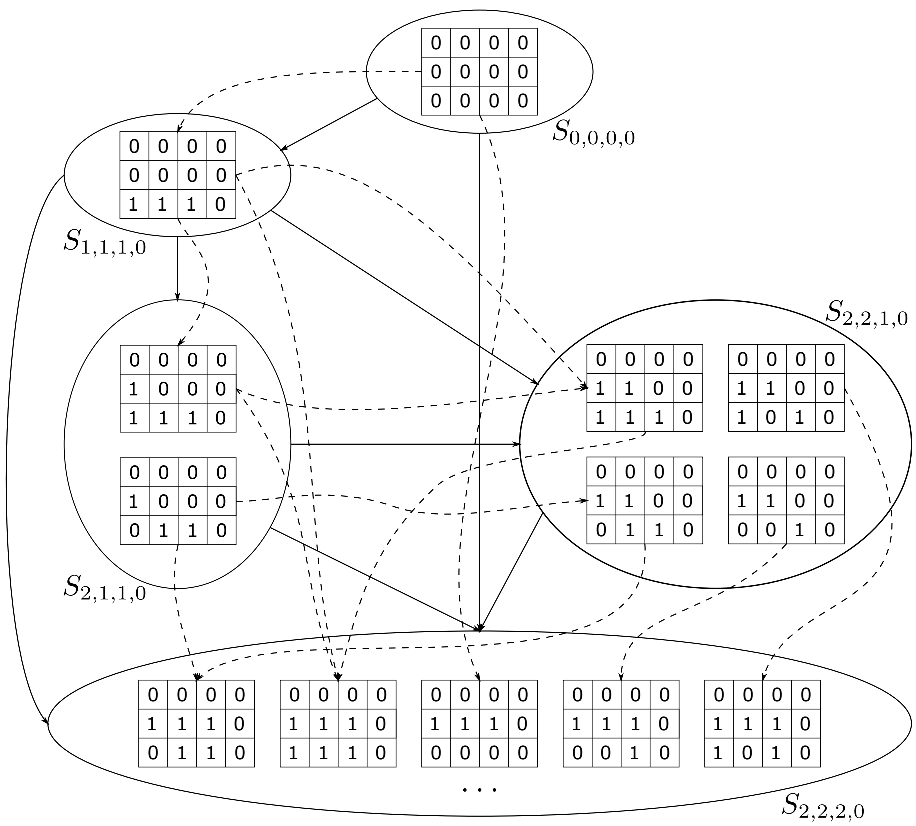

FIGURE 12. Some positions, moves and classes in Auxie with $k = 4$ and $n = 3$.

(in other words: for each i, if $j_i > 0$, then the last 1 in $(y_1^i, \dots, y_n^i)$ is at the position j_i, while if $j_i = 0$, then $(y_1^i, \dots, y_n^i) = e_0$). Let

$$\mathcal{P}' = \{S_{j_1,j_2,\dots,j_k} : n \geqslant j_1 \geqslant j_2 \geqslant \dots \geqslant j_k \geqslant 0\}.$$

Then $\mathcal{P}'$ is a congruential partition of Auxie.

Proof. a) Assume $(e_{j_1}, e_{j_2}, \dots, e_{j_k}) \in P'$. Let $e_{j_i} = (x_1^i, \dots, x_n^i)$ for $1 \leqslant i \leqslant k$. We have that, whenever $m \geqslant j_i + 1$, then $x_m^i = 0$; therefore, the condition (3.2) implies that for each i', $i' > i$, we have $x_m^{i'} = 0$ whenever $m \geqslant j_i + 1$, but this gives $j_{i'} \leqslant j_i$, which was to be proved. The other direction is similar.

b) Clearly, $\mathcal{P}'$ is a partition of P' (indeed, each member of P' belongs to $S_{j_1,j_2,\dots,j_k}$ where j_i marks the position of the last 1 in the i^{th} element of the considered member of P', and $j_i = 0$ if the i^{th} element is $(0, \dots, 0)$). We are left to prove the condition (2.1).

Let $((z_1^1, \dots, z_n^1), \dots, (z_1^k, \dots, z_n^k)) \in f'(((w_1^1, \dots, w_n^1), \dots, (w_1^k, \dots, w_n^k)))$. Let

$$((w_1^1, \dots, w_n^1), \dots, (w_1^k, \dots, w_n^k)) \in S_{j_1,j_2,\dots,j_k},$$
$$((z_1^1, \dots, z_n^1), \dots, (z_1^k, \dots, z_n^k)) \in S_{j'_1,j'_2,\dots,j'_k}.$$

From (3.3) we see that, for $u \leqslant i \leqslant u + v$ (where u and v are as chosen there), we have $j_i < j'_i = t$, for $i < u$ we have $j_i = j'_i \geqslant t$, while for $i > u + v$ we have $j_i = j'_i < t$. Let now $((x_1^1, \dots, x_n^1), \dots, (x_1^k, \dots, x_n^k))$ be any member of $S_{j_1,j_2,\dots,j_k}$. Let us choose the same u, v and t, and define $((y_1^1, \dots, y_n^1), \dots, (y_1^k, \dots, y_n^k))$ as in (3.3). Then

$$((y_1^1, \dots, y_n^1), \dots, (y_1^k, \dots, y_n^k)) \in f'(((x_1^1, \dots, x_n^1), \dots, (x_1^k, \dots, x_n^k)))$$

and

$$((y_1^1, \dots, y_n^1), \dots, (y_1^k, \dots, y_n^k)) \in S_{j'_1,j'_2,\dots,j'_k},$$

which was to be proved. □

We are now ready to prove the following theorem.

Theorem 3.17. *Chomp and Auxie are emulationally equivalent.*

Proof. Let $\mathcal{P}$ be the trivial partition of Chomp, and $\mathcal{P}'$ the partition of Auxie from the previous proposition. We define the bijection $F : \mathcal{P} \to \mathcal{P}'$ by

$$F(\{(a_1, a_2, \dots, a_k)\}) = S_{a_1,a_2,\dots,a_k}.$$

Since $(a_1, a_2, \dots, a_k)$ is nonincreasing, Proposition 3.16a) gives that F is well-defined and that it is surjective, and it is also clear that F is injective. We have $F(\{(0, \dots, 0)\}) = S_{0,\dots,0}$, and Proposition 3.16b) gives $S_{0,\dots,0} = \{b_0\}$. Therefore, we are left to prove the condition (2.2).

($\Rightarrow$): Let $(a_1^1, \dots, a_k^1), (a_1^2, \dots, a_k^2) \in P$ be such that

$$(a_1^2, \dots, a_k^2) \in f((a_1^1, \dots, a_k^1)).$$

Note that, by the definition of f, this means that there exist t, u, v, where $t \in \{1, \ldots, n\}, u \in \{1, \ldots, k\}, v \in \{0, 1, \ldots, k-1\}$, such that:

$$\begin{aligned} &a_i^1 = a_i^2, \text{ for } i < u; \\ &a_i^1 < a_i^2 = t, \text{ for } u \leqslant i \leqslant u+v; \\ &a_i^1 = a_i^2, \text{ for } i > u+v. \end{aligned} \tag{3.4}$$

We need to find a member of $S_{a_1^1, a_2^1, \ldots, a_k^1}$ whose image under f' contains a member of $S_{a_1^2, a_2^2, \ldots, a_k^2}$. We claim that $(e_{a_1^1}, e_{a_2^1}, \ldots, e_{a_k^1})$ is such a member. And indeed, if u, v and t are as in (3.4), then the transformation (3.3) (where it is easily seen that the necessary condition above (3.3) holds) leads to a member of $S_{a_1^2, a_2^2, \ldots, a_k^2}$, which was to be proved.

($\Leftarrow$): Now, let

$$S_{a_1^2, a_2^2, \ldots, a_k^2} \in f'(S_{a_1^1, a_2^1, \ldots, a_k^1}). \tag{3.5}$$

We want to prove that

$$(a_1^2, \ldots, a_k^2) \in f((a_1^1, \ldots, a_k^1)). \tag{3.6}$$

Recall that $a_1^1, a_2^1, \ldots, a_k^1$ mark positions of the last 1s in n-tuples that compose any k-tuple from $S_{a_1^1, a_2^1, \ldots, a_k^1}$, and similarly for $a_1^2, a_2^2, \ldots, a_k^2$. Because of (3.5), we may choose u and v as in (3.3). The requirement preceding (3.3) gives $a_u^1, a_{u+1}^1, \ldots, a_{u+v}^1 < t$, and after applying f' we conclude $a_u^2 = a_{u+1}^2 = \cdots = a_{u+v}^2 = t$, as well as $a_i^1 = a_i^2$ whenever $i < u$ or $i > u+v$. This immediately gives (3.6), which completes the proof. $\square$

3.4.2. *Auxie and Hackenforb.* We shall now show that Auxie is emulationally equivalent to an instance of Hackenforb. Let us roughly describe the main idea. The starting position in Hackenforb will be a graph obtained by taking n paths and identifying one end of each of them with a common vertex (similar to what is shown in Figure 7). If the paths are of lengths $a_1, a_2, \ldots, a_n$, we shall denote such graph by $\circledast(a_1, a_2, \ldots, a_n)$; by convention, we shall always assume $a_1 \leqslant a_2 \leqslant \cdots \leqslant a_n$.

Until the end of this section, we shall assume, for technical reasons, $n \geqslant 3$. (Note that this does not affect the generality of our result. Indeed, if we show that Chomp on some initial board is emulationally equivalent to an instance of Hackenforb, then the same holds for Chomp on any smaller board, since that smaller board appears as a position in the game played on the larger board.) The starting position G in our instance of Hackenforb will be the graph $\circledast(b_1, b_2, \ldots, b_n)$, where

$$b_j = 1 + (2^{j+1} - j - 2) \sum_{u=0}^{k-1} 2^{u(n-1)} \tag{3.7}$$

for $j \in \{1, \ldots, n-1\}$, and

$$b_n = 1 + (2^{n+1} - n - 2) \sum_{u=0}^{k-1} 2^{u(n-1)} - 2^{k(n-1)}. \tag{3.8}$$

(Obviously, $b_j < b_{j'}$ whenever $j < j'$.) Let us now describe the other possible positions in Hackenforb. We first need some preparatory work. Each position in Auxie that is not an ending position will correspond to one position in Hackenforb. Recall that each position in Auxie (a k-tuple of n-tuples) can be represented as a rectangular board with n rows and k columns, filled by 0s and 1s. To each field, say (i, j) (which is the field at the intersection of the i^{th} column and the j^{th} row), we shall assign the value $2^{(i-1)(n-1)+j-1}$, with the exception of the field (k, n), whose assigned value will be 0 (instead of $2^{k(n-1)}$). That way, the bottom left field has the assigned value 1, each field from the second row upwards has double the value of the field below it, and the bottom field in the next column has the same value as the top field in the previous column; the only exception to these rules is the top-right field (see Figure 13). Note that b_1 is the sum of values in the bottom row increased by 1, b_2 is b_1 increased by the total sum of values in the bottom 2 rows, b_3 is b_2 increased by the total sum of values in the bottom 3 rows etc.

2^3	2^6	2^9	2^{12}	0
2^2	2^5	2^8	2^{11}	2^{14}
2^1	2^4	2^7	2^{10}	2^{13}
1	2^3	2^6	2^9	2^{12}

FIGURE 13. Field values with $k = 5$ and $n = 4$.

Let $((x_1^1, \ldots, x_n^1), \ldots, (x_1^k, \ldots, x_{n-1}^k, 0))$ be a fixed (non-ending) position in Auxie. We define:

$$c_j = \sum_{i=1}^{k} \left(x_j^i \sum_{t=1}^{j} x_t^i 2^{(i-1)(n-1)+t-1} \right) \tag{3.9}$$

for $j \in \{1, \ldots, n\}$. In other words: for the j^{th} row, we make note of all the 1s in that row, and for each of them, we sum the values of the field it stands on, as well as all the fields below it that have 1s on them; c_j is the sum of all these values.

Let $T = \sum_{u=0}^{k-1} 2^{u(n-1)}$ (which is the sum of all the values in the bottom row). We have

$$0 \leqslant c_j \leqslant \sum_{i=1}^{k} \left(2^{(i-1)(n-1)} \sum_{t=1}^{j} 2^{t-1} \right) = (2^j - 1)T$$

for $j < n$, and

$$0 \leqslant c_n \leqslant (2^n - 1)T - 2^{k(n-1)}.$$

Finally, the graph corresponding to the considered position from Auxie shall be $✷(b_1 - c_1, \ldots, b_n - c_n)$. We are left to check whether these arguments are in nondecreasing order. And indeed:

$$\begin{aligned} b_j - c_j &\geqslant 1 + (2^{j+1} - j - 2)T - (2^j - 1)T \\ &= 1 + (2^j - (j-1) - 2)T \\ &= b_{j-1} \\ &\geqslant b_{j-1} - c_{j-1} \end{aligned}$$

for all $j < n$, and

$$\begin{aligned} b_n - c_n &\geqslant 1 + (2^{n+1} - n - 2)T - 2^{k(n-1)} - ((2^n - 1)T - 2^{k(n-1)}) \\ &= 1 + (2^n - (n-1) - 2)T \\ &= b_{n-1} \\ &\geqslant b_{n-1} - c_{n-1}. \end{aligned}$$

Let A be the set of all graphs obtained this way (from all the possible non-ending positions of Auxie), and let $\mathcal{F}$ be the set of all connected components of all subgraphs of G that are not in A. We have thus defined our instance of Hackenforb. We write $\Gamma(G, \mathcal{F}) = (P^*, f^*, G)$ (note that $P^* = A \cup \{K_0\}$). For $k = 4$ and $n = 3$, some possible positions in the defined instance of Hackenforb and moves between them are shown in Figure 14 (these are precisely the positions corresponding to the Auxie positions from Figure 12).

The idea of emulation is the following one. In a Hackenforb position $✷(a_1, a_2, \ldots, a_n)$, the path of length a_j corresponds to the j^{th} row in an Auxie position. As it will be shown, the length of such a path will encode one of the possible arrangements of 0s and 1s in the corresponding Auxie row. A move in Auxie, which replaces a horizontal sequence of 0s by 1s, corresponds to a Hackenforb move that removes a number of edges from the corresponding path, in such a way that (reasonably) the resulting path encodes the resulting values in the concerned row after the Auxie move.

We shall now show that the described correspondence between the non-ending positions of Auxie and our graphs is invertible.

Lemma 3.18. *Let $a_1, a_2, \ldots, a_n$ be such that $✷(a_1, \ldots, a_n) \in P^*$. Then there exists unique $((x_1^1, \ldots, x_n^1), \ldots, (x_1^k, \ldots, x_{n-1}^k, 0)) \in P'$ whose corresponding graph is $✷(a_1, \ldots, a_n)$.*

Proof. By our definition of P^*, for each graph $✷(a_1, \ldots, a_n)$ from P^* there exists a corresponding non-ending Auxie position. Therefore, we need to prove only the uniqueness. We shall now present an algorithm that, given $(a_1, \ldots, a_n)$, reconstructs all x_j^i, $1 \leqslant i \leqslant k$, $1 \leqslant j \leqslant n$. (Note that, because of the assumption $n \geqslant 3$, the values $a_1, \ldots, a_n$ are indeed known: they are

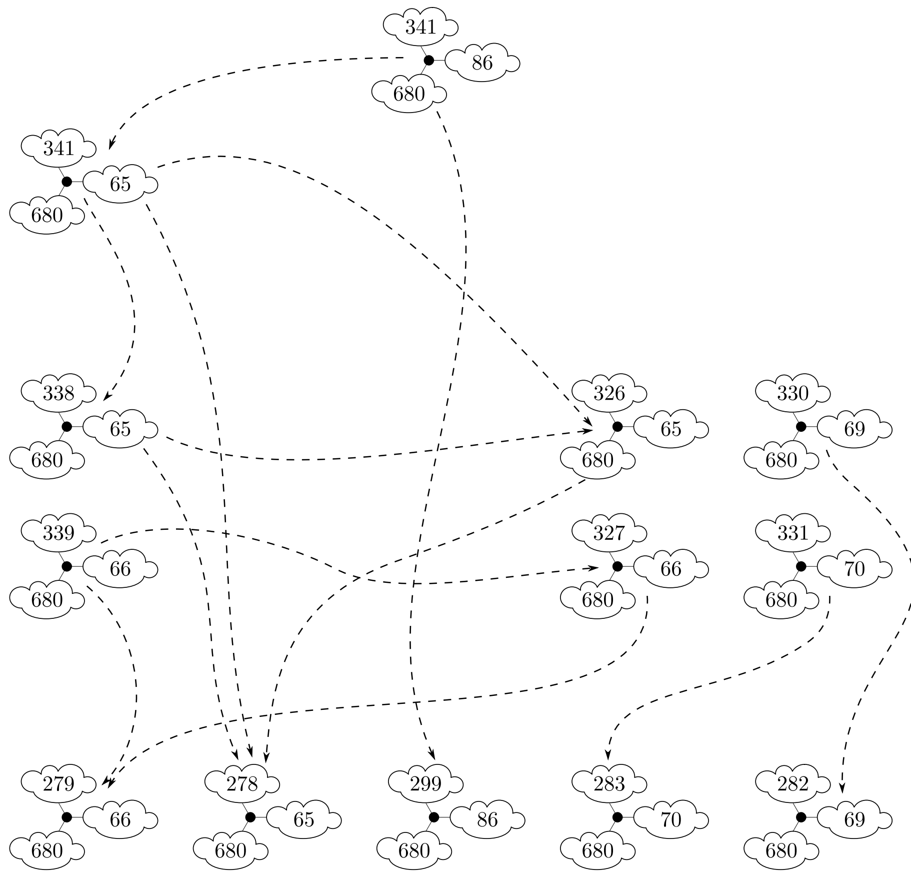

FIGURE 14. Some positions and moves in the defined instance of Hackenforb with $k = 4$ and $n = 3$.

lengths of paths starting from the central vertex, which is the only vertex of degree more than 2.)

We first reconstruct the sequence $(c_1, \ldots, c_n)$, which is easy: $(c_1, \ldots, c_n) = (b_1, \ldots, b_n) - (a_1, \ldots, a_n)$ (where b_i's are as given in (3.7) and (3.8)). Now, since

$$c_1 = \sum_{i=1}^{k} \left(x_1^i \sum_{t=1}^{1} x_t^i 2^{(i-1)(n-1)+t-1} \right) = \sum_{i=1}^{k} (x_1^i)^2 2^{(i-1)(n-1)},$$

and since all the powers of 2 under the sum are different, the binary expansion of c_1 uniquely determines x_1^i for $1 \leqslant i \leqslant k$. A similar idea works for any j whenever $j < n$: indeed, all the values assigned to the fields in the j^{th} row are powers of 2 whose exponent is congruent to $j - 1$ modulo $n - 1$; therefore, in order to reconstruct x_j^i, by (3.9) we see that $x_j^i = 1$ if and only if $2^{(i-1)(n-1)+j-1}$ appears as a summand in the binary expansion of c_j.

Finally, we need to reconstruct the values x_n^i, $1 \leqslant i < k$. Note that each Auxie position has the property that, if $x_n^{i_0} = 1$ for some i_0, then $x_n^i = 1$ whenever $i < i_0$ (which follows from (3.2) for $l = n$). Therefore, we only need to prove that there cannot exist two vectors of the form $(1, \ldots, 1, 0, \ldots, 0)$ (or possibly all 0s) that could stand for $(x_n^1, \ldots, x_n^k)$. Suppose the contrary: assume that there exist two such vectors, with exactly i_0, respectively i_0' 1s, $0 \leqslant i_0 < i_0' < n$. By the definition of c_n, we have

$$\begin{aligned}&\sum_{i=1}^{i_0}\left(\sum_{t=1}^{n-1} x_t^i 2^{(i-1)(n-1)+t-1} + 2^{i(n-1)}\right)\\ =&\sum_{i=1}^{i_0'}\left(\sum_{t=1}^{n-1} x_t^i 2^{(i-1)(n-1)+t-1} + 2^{i(n-1)}\right)\end{aligned}$$

(namely, our assumption is that c_n equals both these values; also note, we use the fact that the values x_j^i for $j < n$ and any i are uniquely determined). The above equality reduces to

$$0 = \sum_{i=i_0+1}^{i_0'}\left(\sum_{t=1}^{n-1} x_t^i 2^{(i-1)(n-1)+t-1} + 2^{i(n-1)}\right),$$

which is clearly a contradiction. This completes the proof. □

We now define a congruential partition of Auxie that we shall use. In short, all the ending positions will be together in one class, and all the other positions will be alone in their class (recall that a similar partition was used for Notakto; see Proposition 3.8).

Proposition 3.19. *Let*

$$\begin{aligned}\mathcal{P}'' = &\Big\{\big\{((x_j^i)_{j=1}^n)_{i=1}^k\big\} : ((x_j^i)_{j=1}^n)_{i=1}^k \in P' \text{ and } x_n^k = 0\Big\}\\ &\cup \Big\{\big\{((x_j^i)_{j=1}^n)_{i=1}^k : ((x_j^i)_{j=1}^n)_{i=1}^k \in P' \text{ and } x_n^k = 1\big\}\Big\}.\end{aligned}$$

Then $\mathcal{P}''$ is a congruential partition of Auxie.

Proof. $\mathcal{P}''$ is clearly a partition of P'. Let us first show that a position $((x_j^i)_{j=1}^n)_{i=1}^k$ in Auxie is an ending position if and only if $x_n^k = 1$.

Assume first $x_n^k = 1$. We need to prove that there do not exist u, v and t from the second point in Definition 3.15. We have already seen that, for each Auxie position, we have that $(x_n^1, x_n^2, \ldots, x_n^k)$ is of the form $(1, \ldots, 1, 0, \ldots, 0)$; therefore, if $x_n^k = 1$, then $(x_n^1, x_n^2, \ldots, x_n^k) = (1, 1, \ldots, 1)$, but then the condition above (3.3) clearly cannot be satisfied for any u, v, t.

Assume now $x_n^k = 0$. Let $(x_n^1, x_n^2, \ldots, x_n^k) = (1, \ldots, 1, 0, \ldots, 0)$, where there are exactly l 1s ($l < k$). Then we can take $u = l+1$, $v = k-1-l$ and $t = n$. The position obtained from $((x_j^i)_{j=1}^n)_{i=1}^k$ for this choice of parameters (by (3.3), where the necessary condition is clearly fulfilled) is a valid position

((3.2) is trivially true, since $y_n^i = 1$ for each i, and thus the left-hand side of the implication is never true).

This proves our claim. Because of that claim, for the rest of the proof we may now proceed as in the proof of Proposition 3.8. □

We are now ready for the main theorem in this section.

Theorem 3.20. *Auxie is emulationally equivalent to an instance of Hackenforb.*

Proof. Let $\mathcal{P}''$ be the partition of Auxie from the previous proposition, and $\mathcal{P}^*$ the trivial partition of Hackenforb. Define $F : \mathcal{P}'' \to \mathcal{P}^*$ by

$$F(\{((x_1^1, \dots, x_n^1), \dots, (x_1^k, \dots, x_{n-1}^k, 0))\}) = \{✺(b_1 - c_1, \dots, b_n - c_n)\},$$

where b_j and c_j are defined by (3.7), (3.8), (3.9), and

$$F\big(\big\{((x_j^i)_{j=1}^n)_{i=1}^k : ((x_j^i)_{j=1}^n)_{i=1}^k \in P' \text{ and } x_n^k = 1\big\}\big) = \{K_0\}.$$

By the definition of P^* and Lemma 3.18, F is bijective. By definition, $F(\{((0, \dots, 0), \dots, (0, \dots, 0))\}) = \{✺(b_1, \dots, b_n)\} = \{G\}$. We are left to prove the condition (2.2).

The direction (⇒). Let

$$((y_1^1, \dots, y_n^1), \dots, (y_1^k, \dots, y_n^k)) \in f'(((x_1^1, \dots, x_n^1), \dots, (x_1^k, \dots, x_n^k))).$$

Then $x_n^k = 0$. Let us write

$$F(\{((x_1^1, \dots, x_n^1), \dots, (x_1^k, \dots, x_{n-1}^k, 0))\}) = \{✺(b_1 - c_1, \dots, b_n - c_n)\}.$$

We shall distinguish the cases $y_n^k = 0$ and $y_n^k = 1$. Let first $y_n^k = 0$. Let us write

$$F(\{((y_1^1, \dots, y_n^1), \dots, (y_1^k, \dots, y_{n-1}^k, 0))\}) = \{✺(b_1 - d_1, \dots, b_n - d_n)\}.$$

By the definition of f', there exist $u \in \{1, \dots, k\}$, $v \in \{0, \dots, k-1\}$ and $t \in \{1, \dots, n\}$ such that:

$$\begin{gathered} x_j^i = y_j^i, \text{ for } j \neq t; \\ x_j^i = y_j^i = 0, \text{ for } j > t \text{ and } i \geqslant u; \\ x_t^i = y_t^i = 0, \text{ for } i > u + v; \\ x_t^i = 0,\ y_t^i = 1, \text{ for } u \leqslant i \leqslant u + v; \\ x_t^i = y_t^i, \text{ for } i < u. \end{gathered} \tag{3.10}$$

From (3.10), we have:

$$\begin{aligned} d_j &= c_j \text{ for } j \neq t; \\ d_t &= \sum_{i=u}^{u+v} \left(\sum_{s=1}^{t} y_s^i 2^{(i-1)(n-1)+s-1} \right) + c_t. \end{aligned}$$

Therefore, the only difference between the graphs $✺(b_1 - c_1, \dots, b_n - c_n)$ and $✺(b_1 - d_1, \dots, b_n - d_n)$ is that one path is shorter in the second one

$(b_t - c_t > b_t - d_t)$, which means that there is a move from the first one to the second one.

Assume now $y_n^k = 1$. Then, since the obtained position is an ending position, we need to show that, in this case, $K_0 \in f^*(\circledast(b_1-c_1,\dots,b_n-c_n))$. But this is easy: removing, for example, any edge incident to the central vertex leads to K_0, which was to be proved.

The direction ($\Leftarrow$). Let

$$\circledast(b_1 - d_1, \dots, b_n - d_n) \in f^*(\circledast(b_1 - c_1, \dots, b_n - c_n)) \tag{3.11}$$

(where both those graphs are in P^*; note that the expression above implicitly assumes that we make a move to a non-ending position, because the case with K_0 on the left-hand side will be treated at the end), and

$$F(\{((x_1^1,\dots,x_n^1),\dots,(x_1^k,\dots,x_n^k))\}) = \{\circledast(b_1 - c_1,\dots,b_n - c_n)\},$$
$$F(\{((y_1^1,\dots,y_n^1),\dots,(y_1^k,\dots,y_n^k))\}) = \{\circledast(b_1 - d_1,\dots,b_n - d_n)\}.$$

By the definition of F, we have $x_n^k = y_n^k = 0$. The aim is to prove

$$((y_1^1,\dots,y_n^1),\dots,(y_1^k,\dots,y_n^k)) \in f'(((x_1^1,\dots,x_n^1),\dots,(x_1^k,\dots,x_n^k))). \tag{3.12}$$

From (3.11), it follows that there exists $t \in \{1,\dots,n\}$ such that

$$\begin{aligned} b_j - c_j &= b_j - d_j \text{ for } j \neq t; \\ b_t - c_t &> b_t - d_t, \end{aligned} \tag{3.13}$$

which implies $c_j = d_j$ for $j \neq t$, and $c_t < d_t$.

We first treat the case $t = n$. Since, in this case, $c_j = d_j$ whenever $j < n$, we have $x_j^i = y_j^i$ for $j < n$ and i arbitrary (as we have seen in the proof of Lemma 3.18). We have also already seen that both $(x_n^1,\dots,x_n^k)$ and $(y_n^1,\dots,y_n^k)$ have to be of the form $(1,\dots,1,0,\dots,0)$; say with exactly l, respectively m 1s, where $0 \leqslant l < m < k$ (because of $c_n < d_n$). Therefore, in the definition of f' we can choose $t = n$, $u = l + 1$, $v = m - l - 1$, which gives (3.12).

Assume now $t < n$. Recall that, as seen during the proof of Lemma 3.18, the value c_j for any fixed j, $j < n$, is solely enough to uniquely determine all the values x_j^i. Therefore, since $c_j = d_j$ whenever $j \neq t$, we conclude $x_j^i = y_j^i$ for $t \neq j < n$ and i arbitrary. Let us also show that $x_n^i = y_n^i$ for any i. Assume that $(x_n^1,\dots,x_n^k)$ contains exactly l 1s (in the first l positions), and $(y_n^1,\dots,y_n^k)$ exactly m 1s. If, say, $l = 0$, then $c_n = 0$, and then $d_n = c_n = 0$, which implies $m = 0$; therefore, $x_n^i = y_n^i (= 0)$ for any i, as claimed. Let now

$l, m > 0$. Assume first $l > m$. We have:

$$\begin{aligned} d_n &= \sum_{i=1}^{m}\sum_{s=1}^{n} y_s^i 2^{(i-1)(n-1)+s-1} \\ &= \sum_{i=1}^{m}\sum_{s=1}^{n-1} y_s^i 2^{(i-1)(n-1)+s-1} + \sum_{i=1}^{m} 2^{i(n-1)} \\ &\leqslant \sum_{i=1}^{l}\sum_{s=1}^{n-1} 2^{(i-1)(n-1)+s-1} + \sum_{i=1}^{l-1} 2^{i(n-1)} \\ &= \sum_{z=0}^{(l-1)(n-1)+n-2} 2^z + \sum_{i=1}^{l-1} 2^{i(n-1)} \\ &< 2^{l(n-1)} + \sum_{i=1}^{l-1} 2^{i(n-1)} = \sum_{i=1}^{l} 2^{i(n-1)} \leqslant c_n, \end{aligned}$$

which contradicts $d_n = c_n$. In an analogous way, we show that the assumption $l < m$ implies $c_n < d_n$. Therefore, the only possibility is $l = m$, which gives $x_n^i = y_n^i$ for any i, as claimed. Let us also show that, for $i \leqslant l$ and $j \leqslant n-1$, we have $x_j^i = y_j^i$ (actually, only the case $j = t$ is interesting here, since otherwise we already have that equality). We have:

$$\begin{aligned} c_n - \sum_{i=1}^{l} 2^{i(n-1)} &= \sum_{i=1}^{l}\sum_{s=1}^{n-1} x_s^i 2^{(i-1)(n-1)+s-1}; \\ d_n - \sum_{i=1}^{l} 2^{i(n-1)} &= \sum_{i=1}^{l}\sum_{s=1}^{n-1} y_s^i 2^{(i-1)(n-1)+s-1}; \end{aligned}$$

therefore, since the left-hand sides are equal, the same must be true for the right-hand sides. But note that the right-hand sides are sums of different powers of 2, which gives $x_j^i = y_j^i$ whenever $i \leqslant l$ and $j \leqslant n-1$, as claimed.

To sum up, we know so far that, if $x_j^i \neq y_j^i$, then $j = t$ and $i > l$. What is left to prove (in order to show (3.12)) is that: 1) the values i such that $x_t^i \neq y_t^i$ are consecutive; 2) for each such i, we have $x_t^i = 0$ and $y_t^i = 1$ (not vice versa); 3) if u is the minimal such i, then $x_j^m = 0$ whenever $m \geqslant u$ and $j \geqslant t$. Let us show these statements.

2), 3): Let u be as described in 3). We have $u > l$. We show first $x_t^u = 0$ and $y_t^u = 1$. Suppose the contrary: $x_t^u = 1$ and $y_t^u = 0$. We show that $y_j^u = 0$ whenever $j \geqslant t$. Indeed, we already know $y_n^u = 0$ (since $u > l$). Suppose that $y_{j_0}^u = 1$ for some j_0, $t < j_0 < n$; then also $x_{j_0}^u = y_{j_0}^u = 1$. Recall that, if $y_{j_0}^u = 1$, then the value d_{j_0} is solely enough to uniquely determine the values y_j^u for all $j \leqslant j_0$ (this can be seen by the same arguments as used in the proof of Lemma 3.18). Therefore, since $c_{j_0} = d_{j_0}$ and $x_{j_0}^u = y_{j_0}^u = 1$, it follows that $x_t^u = y_t^u$, which is a contradiction. Therefore, $y_j^u = 0$ whenever $j \geqslant t$. Since $((y_j^i)_{j=1}^n)_{i=1}^k \in P'$, (3.2) gives $y_j^m = 0$ whenever $m > u$ and $j \geqslant t$, and in particular, $y_t^m = 0$ whenever $m > u$; together with $x_t^u = 1$ and $y_t^u = 0$, this implies $c_t > d_t$, contradicting (3.13). Therefore, $x_t^u = 0$ and $y_t^u = 1$. Now we prove $x_j^m = 0$ whenever $m > u$ and $j \geqslant t$ in exactly the same way as we have just seen, which gives 3), while 2) directly follows from 3).

1): If $x_t^i \neq y_t^i$ for all $i \geqslant u$, the claim is true. Assume now that v is a nonnegative integer such that $x_t^i \neq y_t^i$ whenever $u \leqslant i \leqslant u+v$, and $x_t^{u+v+1} = y_t^{u+v+1}$. Because of the part 3), we conclude $y_t^{u+v+1} = x_t^{u+v+1} = 0$. Now the same argument as in the previous paragraph gives $y_t^i = x_t^i = 0$ whenever $i \geqslant u+v+1$. Therefore, the values i such that $x_t^i \neq y_t^i$ are exactly $u, u+1, \ldots, u+v$, which was to be proved.

This concludes the proof that (3.11) implies (3.12). Finally, we need to take care of the postponed case

$$K_0 \in f^*(\text{✷}(b_1 - c_1, \ldots, b_n - c_n)),$$

where

$$F(\{((x_1^1, \ldots, x_n^1), \ldots, (x_1^k, \ldots, x_n^k))\}) = \{\text{✷}(b_1 - c_1, \ldots, b_n - c_n)\}.$$

We need to prove that there is a move from a corresponding Auxie position to an ending position. But this is easy: we know that $(x_n^1, \ldots, x_n^k) = (1, \ldots, 1, 0, \ldots, 0)$ (with possibly no 1s), and we can in one move put 1s in all the positions x_n^i in which there are 0s, and thus obtain an ending position. This completes the proof. $\square$

4. Hackenforb is not almighty

The game Halving Nim is a variant of Nim defined as follows. In the beginning, there is a heap of n coins. On each move, if there are t coins left on the heap, a player can take any number of coins between 1 and $\lceil \frac{t}{2} \rceil$ (inclusively). The game ends when there are no coins left.

We shall prove that this game is not emulationally equivalent to an instance of Hackenforb. In particular, we shall show this for the case $n = 3$ (and thus also for all larger values of n).

Definition 4.1. *The considered instance of the Halving Nim is defined as the triple $(P, f, 3)$, where $P = \{0, 1, 2, 3\}$, and $f : P \to \mathcal{P}(P)$ is given by*

$$f(0) = \emptyset, \ f(1) = \{0\}, \ f(2) = \{1\}, \ f(3) = \{1, 2\}.$$

All the possible moves in this game are shown in Figure 15.

Let us now prove that the only congruential partition of this game is the trivial one.

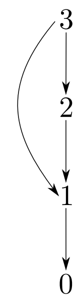

Figure 15. Possible moves for Halving Nim with 3 coins.

Proposition 4.2. *The only congruential partition of the game* $(P, f, 3)$ *is given by*

$$\mathcal{P} = \{\{0\}, \{1\}, \{2\}, \{3\}\}.$$

Proof. Since the position 0 is the only one from which there are no possible moves, it has to be alone in its class (otherwise (2.1) would not hold). We now show that no two of the remaining three positions can be in the same class.

- If the positions 3 and 2 are in the same class, say A, then, since $f(3) \ni 2 \in A$, there must exist $a \in A$ such that $a \in f(2)$, which directly implies $1 \in A$. Now, using $f(1) = \{0\}$, the same reasoning gives $0 \in A$, which is impossible since 0 is in its own class.
- If the positions 3 and 1 are in the same class, then 0 is also in that class (for the same reason as above), a contradiction.
- Finally, if 1 and 2 are in the same class, then again 0 is in it, too, a contradiction.

This completes the proof. □

We are now ready for the main theorem of this part.

Theorem 4.3. *Halving Nim for* $n = 3$ *is not emulationally equivalent to any instance of Hackenforb.*

Proof. Suppose the contrary: there exists a graph G and a set $\mathcal{F}$ of forbidden graphs such that Halving Nim for $n = 3$ is emulationally equivalent to $\Gamma(G, \mathcal{F})$; let $\Gamma(G, \mathcal{F}) = (P^*, f^*, G)$. Let $\mathcal{P}^*$ be a congruential partition of $\Gamma(G, \mathcal{F})$ that corresponds to the partition $\mathcal{P}$; we may write $\mathcal{P}^* = \{\mathcal{P}_0^*, \mathcal{P}_1^*, \mathcal{P}_2^*, \mathcal{P}_3^*\}$ with the natural correspondence between $\mathcal{P}$ and $\mathcal{P}^*$ (in particular, $\mathcal{P}_3^* = \{G\}$ and $\mathcal{P}_0^* = \{K_0\}$).

While playing Hackenforb, we shall, for the purpose of this proof, keep track of the forbidden components that appear during the game; instead of erasing them, we shall leave them on the "playing board," but (of course) they cannot be used in the rest of the game. Edges that are not in forbidden components will be called *allowed.*

Removing any edge of G results in a position either from $\mathcal{P}_2^*$ or from $\mathcal{P}_1^*$. Based on that distinction, edges will be called of *type 1* or *type 2*, respectively. Let T_1, respectively T_2, be the set of all edges of type 1, respectively type 2. The removal of any edge of type 2 leads to a position such that any further removal of any allowed edge in that position results in a graph whose all connected components are forbidden. On the other hand, since each move from a position in $\mathcal{P}_2^*$ leads to a position from $\mathcal{P}_1^*$, we conclude that the removal of any edge of type 1 leads to a position such that any further removal of an allowed edge in the remaining position results in a graph with at least one connected component that is not forbidden. In brief:

$$e \in T_1 \Leftrightarrow (\forall g \in G - e)(g \text{ allowed in } G - e \Rightarrow (\exists G' \in C(G - e - g))(G' \notin \mathcal{F}));$$

$$e \in T_2 \Leftrightarrow (\forall g \in G - e)(g \text{ allowed in } G - e \Rightarrow (\forall G' \in C(G - e - g))(G' \in \mathcal{F})).$$

(Here $C(\dots)$ denotes the set of all connected components of a given graph.)

In order to reach a contradiction, it will be enough to find two edges $f \in T_1$ and $h \in T_2$ such that f is allowed in $G - h$ and h is allowed in $G - f$. Indeed, in that case, since $h \in T_2$, all the connected components of $(G - h) - f$ must be forbidden, but since $f \in T_1$, there must exist a connected component of $(G - f) - h$ that is not forbidden; however, this is clearly impossible since $(G - h) - f = (G - f) - h$.

If G is not connected, we can always take one edge from one connected component and another edge from another one in such a way that those two edges are of different types; then each of them will be allowed after removal of the other one, as needed. Therefore, we can assume that G is connected. We say that an edge e generates a forbidden component iff $G - e$ has at least one forbidden component with at least two vertices. If there are edges $f \in T_1$ and $g \in T_2$ none of which generates a forbidden component, then we again have what was needed. Therefore, from now on we assume that each edge of type 1 generates a forbidden component or each edge of type 2 generates a forbidden component.

Assume first that each edge of type 2 generates a forbidden component. Let $h \in T_2$ and $G - h = C \cup F$, where C and F are two connected components of $G - h$ and $F \in \mathcal{F}$. The component C must be allowed (since the game must not be over at this point). We claim that there is an edge f in the component C such that $C - f$ and $G - f$ are both connected, with the possible exception of an isolated vertex. If C is a tree, then it has at least two pendant edges, and at least one of them is a pendant edge in G, too; we can take f to be that edge. If C is not a tree, then C has a cycle, and we can take f to be any edge from a cycle. Since $G - f$ is connected, and we assumed that each edge of type 2 generates a forbidden component, it follows that f is of type 1. Therefore, f and h are a pair of edges that we needed.

In the case when each edge of type 1 generates a forbidden component, the proof is analogous. The theorem is thus proved. $\square$

5. Conclusion and further research

The purpose of the present work is to introduce the game Hackenforb as a new common "language" for studying many different combinatorial games. We hope that it is not too ambitious to believe that, in some future, this could help in bringing out some new possible directions from which we can approach various open questions on combinatorial games and shed some light on them.

The result from Section 4 could seem as a kind of disillusionment after the earlier sections, but it should not be perceived that way. It is our belief (judging by the fact that some quite different games shown here are emulationally equivalent to an instance of Hackenforb) that Hackenforb has proved itself to be more than versatile enough, and the epiphany learned in Section 4 (that the world is not perfect) actually provides a motivation for some further research questions. Namely, in the ideal case, it would be possible to obtain the complete characterization of games emulationally equivalent to an instance of Hackenforb. If this turns out to be too much to hope for, one could try to find a class of games, as general as possible, all of which are emulationally equivalent to an instance of Hackenforb. Or, with a similar aim in mind, one could try to isolate the core reason why Halving Nim is not emulationally equivalent to an instance of Hackenforb (in terms of some obstacles implied by its structural properties) and thus be able to better understand the class of games not emulationally equivalent to an instance of Hackenforb. These and similar questions form the basis for future work.

References

1. L. Beaudou, P. Coupechoux, A. Dailly, S. Gravier, J. Moncel, A. Parreau, and É. Sopena, *Octal games on graphs: The game 0.33 on subdivided stars and bistars*, Theoret. Comput. Sci. **746** (2018), 19–35.
2. E.R. Berlekamp, J.H. Conway, and R.K. Guy, *Winning Ways for Your Mathematical Plays, Vol. 1 (2nd ed.)*, AK Peters, Wellesley, Massachusetts, 2001.
3. H.L. Bodlaender, D. Kratsch, and S.T. Timmer, *Exact algorithms for Kayles*, Theoret. Comput. Sci. **562** (2015), 165–176.
4. J.H. Conway, *On Numbers and Games*, Academic Press, London, 1976.
5. M. Gardner, *Mathematical games*, Sci. Am. **226** (1972), no. 1, 104–107.
6. A. Guignard and É. Sopena, *Compound Node–Kayles on paths*, Theoret. Comput. Sci. **410** (2009), 2033–2044.
7. F. Harary and R.W. Robinson, *Connect-it games*, College. Math. J. **15** (1984), 411–419.
8. M.A. Huggan and B. Stevens, *Polynomial time graph families for Arc Kayles*, Integers **16** (2016), Article No. A86, 14 pp.
9. M. Kano, *Edge-removing games of star type*, Discrete Math. **151** (1996), 113–119.
10. R.J. Nowakowski, *Unsolved problems in combinatorial games*, Games of No Chance 4, Cambridge University Press, 2015, pp. 279–308.
11. T.E. Plambeck and G. Whitehead, *The secrets of Notakto: Winning at X-only tic-tac-toe*, Recreat. Math. Mag. **2** (2014), 49–54.

12. T.J. Schaefer, *On the complexity of some two-person perfect-information games*, J. Comput. System Sci. **16** (1978), 185–225.
13. F. Schuh, *Spel van delers*, Nieuw Tijdschr. Wisk. **39** (1952), 299–304.
14. D. Zeilberger, *Chomp, recurrences and chaos(?)*, J. Difference Equ. Appl. **10** (2004), 1281–1293.

Department of Mathematics and Informatics, University of Novi Sad, Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia
E-mail address: `bojan.basic@dmi.uns.ac.rs`

Faculty of Sciences and Mathematics, University of Niš, Višegradska 33, 18000 Niš, Serbia
E-mail address: `nikola5000@gmail.com`

Mathematical Institute of the Serbian Academy of Sciences and Arts, Kneza Mihaila 36, 11000 Beograd, Serbia
E-mail address: `danijela.mitrovic@dmi.uns.ac.rs`