Partition games

Antoine Dailly, Eric Duchene, Urban Larsson, Gabrielle Paris

Partition games

Antoine Dailly[†‡*], Éric Duchêne[†*], Urban Larsson[¶], Gabrielle Paris[†*]

[†]Univ Lyon, Université Lyon 1, LIRIS UMR CNRS 5205, F-69621, Lyon, France.
[‡]Instituto de Matemáticas, UNAM Juriquilla, 76230 Querétaro, Mexico.
[¶]National University of Singapore, Singapore.

Abstract

We introduce CUT, the class of 2-player partition games. These are NIM type games, played on a finite number of heaps of beans. The rules are given by a set of positive integers, which specifies the number of allowed splits a player can perform on a single heap. In normal play, the player with the last move wins, and the famous Sprague-Grundy theory provides a solution. We prove that several rulesets have a periodic or an arithmetic periodic Sprague-Grundy sequence (i.e. they can be partitioned into a finite number of arithmetic progressions of the same common difference). This is achieved directly for some infinite classes of games, and moreover we develop a computational testing condition, demonstrated to solve a variety of additional games. Similar results have previously appeared for various classes of games of TAKE-AND-BREAK, for example OCTAL and HEXADECIMAL; see e.g. Winning Ways by Berlekamp, Conway and Guy (1982). In this context, our contribution consists of a systematic study of the subclass 'break-without-take'.

1 Introduction

This work concerns 2-player combinatorial games related to the classical game of NIM, but instead of removing objects, say beans, from heaps, players are requested to partition the existing heaps into smaller heaps, while the total number of beans remain constant. We prove several regularity results on the solutions of such games.

Integer partition theory, related to Ferrer diagrams and Young tableaus, is a classical subject in number theory and combinatorics, dating back to giants such as Lagrange, Goldbach and Euler; it concerns the number of ways you can write a given positive integer as a sum of specified parts. In most generality, to each positive integer $n \in \mathbb{N} = \{1, 2, \ldots\}$, there belongs a number $p(n)$, which counts the unrestricted number of ways this can be done. For example $4 = 3+1 = 2+2 = 2+1+1 = 1+1+1+1$, so $p(4) = 5$. We may index this partition number by saying exactly how many parts is required, and write $p_k(n)$ for the number of partitions of n in exactly k parts. Thus, in our example, $p_2(4) = 2$ and $p_3(4) = 1$. We could also define $p_{k,c} = p_k + p_c$ and so on. The number of partitions can be beautifully expressed via generating functions, where recurrence formulas, congruence relations, and several asymptotic estimates are known, proved more recently by famous number theorists such as Ramanujan, Hardy, Rademacher and Erdős in the early 1900s. About the same time, a theory of combinatorial games was emerging, via contributions by Bouton, Sprague and Grundy and others, seemingly unrelated to the full blossom of number theory.

An *integer partition game* can be defined by 2 players alternating turns and by specifying the legal partitions, say into exactly 2 or 3 parts, until the current player cannot find a legal partition of parts, and loses. Thus, from position 4, then $3+1, 2+2, 2+1+1$ are the legal move options—if you play to $2+2$ you win, and otherwise not. It turns out that the idea for how to win such games is coded in a 'game function', discovered independently by the mathematicians Sprague and Grundy, which, by the way, does not appear to have any direct relation to the partition function. For example, the partition functions are nondecreasing, but if a Sprague-Grundy function is nondecreasing the game is usually rather trivial, such as the game of NIM on one heap.

Let us begin by giving the relevant game theory background to our results, that several *partition games* have either a periodic or an *arithmetic-periodic* Sprague-Grundy sequence.[1] We call the class of partition games CUT.

An impartial combinatorial game $G = G(\mathcal{R}, X, x)$ is given by a ruleset $\mathcal{R}$, a set of positions X, and a starting position $x \in X$; the ruleset $\mathcal{R}$ specifies how to move from any (starting) position. Two players alternate in moving, with one of the players assigned as a starting player, and in normal play a player who cannot move loses. An *option* of G is a game $G' = G'(\mathcal{R}, X, y)$ that can be reached in one move, and all games have finitely many options, and finite rank.

The *Sprague-Grundy value* [16, 10] (or $\mathcal{G}$-value for short) of an impartial normal play game G is the nonnegative integer

$$\mathcal{G}(G) = \text{mex}\{\mathcal{G}(G') \mid G' \text{ is an option of } G\},$$

where $\text{mex}(U) = \min \mathbb{N}_0 \setminus U$ is the smallest nonnegative integer that does not belong to the strict subset $U \subset \mathbb{N}_0 = \mathbb{N} \cup \{0\}$.

The Sprague-Grundy value can be used to determine the winner in perfect play. Indeed, a game G satisfies $\mathcal{G}(G) = 0$ if and only if playing first in G loses; *i.e.* the *previous player* wins.

Heap games are typically played on several heaps of the form H_n, where $n \in \mathbb{N}_0$ denotes the number of beans in a heap, and there is a given ruleset that specifies the legal options on the heaps. An important concept is that of a disjunctive sum of heaps. A *disjunctive sum* of k heaps of sizes $i_0, \ldots, i_k$ is denoted $H = H_{i_0} + \cdots + H_{i_k}$, where by moving, a player chooses one of the heaps, say H_{i_j} and makes a move in this heap according to a given ruleset. The other heaps remain the same. That is, a typical move option of the game H is of the form H' with

$$H' = H_{i_0} + \cdots + H_{i_{j-1}} + H_{i_j}{}' + H_{i_{j+1}} + \cdots + H_{i_k}, \tag{1}$$

where $H_{i_j}{}'$ is a move option on the heap H_{i_j}.[2]

By the main result of Sprague-Grundy Theory, $\mathcal{G}(H) = \mathcal{G}(H_{i_0}) \oplus \cdots \oplus \mathcal{G}(H_{i_k})$, where $\oplus$ is the standard *nim-sum* operator.[3] Therefore, the previous player wins if and only if the nim-sum is 0.

The following result about the nim-sum operator will be used several times in this paper.

Lemma 1. *Consider any $a_0, \ldots, a_m \in \mathbb{N}_0$. Then*

$$a_0 \oplus a_1 \oplus \cdots \oplus a_m \equiv (a_0 + \cdots + a_m) \bmod 2$$

and

$$a_0 + \cdots + a_m \geq a_0 \oplus \cdots \oplus a_m.$$

Proof. Both the nim-sum and the standard sum is odd if and only if the number of odd a_is is odd. This proves the first part. For each index $j \in [0, m]$, set $\sum_i e_i(j)2^i = a_j$. Then, by definition, for each i, $\sum_j e_j \geq \bigoplus e_j$, which proves the second part. $\qquad\square$

As evidenced by (1), to study a heap game played on a disjunctive sum of heaps, it suffices to describe the rules on a single heap.

Definition 1 (CUT). The CUT *ruleset* $\mathcal{C} \subseteq \mathbb{N}$ specifies the allowed number of *cuts* (or *splits*) of a given heap $H_n = H_n(\mathcal{C})$ into a disjunctive sum of heaps, as

$$H_n = \{H_{n_0} + \cdots + H_{n_c} \mid c \in \mathcal{C}, \forall i : n_i > 0, n_0 + \cdots + n_c = n\}$$

An instance of H_n is called a *c-cut* or an *option* of n, and is denoted $O_n \in H_n$.

---

[1] An arithmetic-periodic sequence can be partitioned into a finite set of arithmetic progressions with the same common difference.

[2] In general, an option may include several heaps, where each new heap is smaller than the previous heap, to allow recursive computation of the Sprague-Grundy values; in this paper, exactly one of the heaps in a sum of heaps will be cut (broken, partitioned, split) into at least 2 heaps. See also Section 4 for a surrounding context and a review of classes of heap games, such as octal games, hexadecimal games, and so on.

[3] Consider nonnegative integers $m = \sum e_i 2^i$ and $n = \sum f_i 2^i$. Then $n \oplus m = \sum (e_i \text{XOR} f_i) 2^i$.

That is, by combining Definition 1 with (1), a move consists in splitting one of the heaps in the disjunctive sum H into $c + 1$ non-empty heaps with $c \in \mathcal{C}$, while the other heaps remain the same.[4]

Let us illustrate Definition 1 with an example. Assume that $\mathcal{C} = \{3, 4\}$, that is, we can split a heap into either 4 or 5 non-empty heaps. Assume that the initial position is the heap H_{13}. The first player decides to split it into four heaps, leaving $H_1 + H_3 + H_3 + H_6$. The second player has to choose one of those four heaps and split it, which they can only do with H_6, after which the game ends since no heap can be split anymore. Note that the first player could have chosen to split the heap into four equal heaps of size 3 and one of size 1, which would have ended the game immediately.

In this paper, we find the $\mathcal{G}$-sequences for several instances of CUT. In Section 2, we solve the classes:

- $\mathcal{C} \subseteq$ 'odd numbers';

- $\min \mathcal{C} \geq 2$;

- $\{1, 2, 3\} \subset \mathcal{C}$;

- $\mathcal{C} = \{1, 3, 2k\}$, k even.

Later, in Section 3, we solve several more instances of CUT by adapting a computational testing condition, that we will call the AP-test, summarized in Table 1 (see also Table 2).

A typical representative of CUT is $G(\mathcal{C}, \mathbb{N}, H)$, but we usually omit $\mathbb{N}$ since it is understood, and also H, because we will gain knowledge of the Sprague-Grundy values of individual heaps (which then implies knowledge about a disjunctive sums of heaps). In this spirit, we continue to write just $G(\mathcal{C})$, and we will return to this notation.

To simplify notation, for the Sprague-Grundy value of a heap H_n of size $n \geq 1$, we will write $\mathcal{G}(H_n) = \mathcal{G}(n)$, and for a disjunctive sum of heaps, we write $H_{n_0} + \cdots + H_{n_k} = (n_0, \ldots, n_k)$ and $\mathcal{G}(H_{n_0} + \cdots + H_{n_k}) = \mathcal{G}(n_0, \ldots, n_k)$.

There are several similarities with NIM-type games, where only removal is possible. For example, if $\mathcal{C} = \mathbb{N}$, then H_{n+1} is equivalent with a NIM heap of size n. This follows, because both are normal play impartial games and $H_n(\text{NIM}) \subset H_{n+1}(\mathcal{C})$, and argue by induction on the Sprague-Grundy values. See also Proposition 17 in Section 4 for a more general observation relating to the classical TAKE-AND-BREAK games.

This work was much inspired by in particular two classical combinatorial games, namely GRUNDY'S GAME [10] and COUPLES-ARE-FOREVER [4]; see also [7]. In the first one, a move consists in choosing a heap and splitting it into two heaps of *different* size. The latter one allows to split any heap of size *at least three* into two heaps. For both games, some extra constraints have been adjoined to the type of possible splits, and no regularities in the $\mathcal{G}$-sequences have yet been observed. Here, we study partition games with no extra constraint than prescribed *splitting*- or *cut*-numbers.

NIM-type games on finite rulesets are known as SUBTRACTION [3], and it is a folklore result that all such games are periodic (i.e. their Sprague-Grundy sequences $(\mathcal{G}(H_n))_{n \in \mathbb{N}}$ are periodic). This follows by a simple combinatorial counting argument. For the game CUT, the situation is more varied, and we will encounter both periodic, and arithmetic-periodic games. Some rulesets are not yet fully understood, and in particular the ruleset $\mathcal{C} = \{1, 2\}$ remains a mystery (it does not seem to be arithmetic-periodic); see Section 5.

We use the notion of a 'game' in at least two different ways, both depending on the notion of a 'ruleset'. A CUT game that can be enjoyed as a recreational game is a ruleset $\mathcal{C}$ together with a heap H_n of a given size n (or a disjunctive sum of heaps H). On the other hand, a minimal requirement for a strategic understanding of a ruleset is to acquire its $\mathcal{G}$-sequence (*i.e.* $\mathcal{G}(H_1), \mathcal{G}(H_2), \ldots$). One consequence of the $\mathcal{G}$-sequence is that you might be indifferent between two rulesets (independently of the size of a heap). From this perspective it is more natural to think of a game, as its sequence of $\mathcal{G}$-values. The context will decide, and we believe that the word "game" should remain a word without a precise definition, available for use in various contexts; on the other hand, a "ruleset" should always have a well defined meaning.

---

[4]Here $\mathcal{C}$ is an *invariant ruleset* in an analogous sense as defined in [6] for subtraction games. The term 'invariant' is with respect to the size of the heap, so that if $c \in \mathcal{C}$ then a c-cut is available for all H_n, with $n > c$. We often think of the sequence of all heap games for a given ruleset $\mathcal{C}$, as the 'fingerprint' of a game. As it is common in CGT papers, we will deemphazise the actual strategies of how to win individual games, and instead evaluate a game in terms of the sequence of Sprague-Grundy values of all heaps.

In this spirit, and to emphasize the motivation of this paper, we will identify $G(\mathcal{C})$ with its sequence of Sprague-Grundy values $\mathcal{G}(H_1), \mathcal{G}(H_2), \ldots$. For example, we write $G(\{1\}) = 0, 1, 0, 1, \ldots$ (See Proposition 3.)

We will index the elements in a given set $\mathcal{C}$ such that, for all i, $c_i < c_{i+1}$.

A well-studied periodicity problem, by Richard Guy, on so-called octal games,[5] does not transfer to the full class of partition games. Indeed, the following lemma establishes that if $\mathcal{C}$ contains an even cut-number, the Sprague-Grundy values are unbounded.

Theorem 2. *If a ruleset $\mathcal{C}$ contains an even cut-number, then $G(\mathcal{C})$ (i.e. its Sprague-Grundy-sequence) is unbounded. In particular, if the smallest even cut-number is $c \in \mathcal{C}$, then any arithmetic progression of the form $x, x+c, x+2c, \ldots$, contains no repetition of $\mathcal{G}$-values, and hence it contains infinitely many $\mathcal{G}$-values.*

Proof. It suffices to show that, for every pair of heaps H_{x_1}, H_{x_2}, with $x_1 \neq x_2$ such that $\mathcal{G}(x_1) = \mathcal{G}(x_2)$, then $x_1 \not\equiv x_2 \bmod c$.

Suppose that $x_1 \equiv x_2 \bmod c$. Then $x_1 = q_1 c + r$ and $x_2 = q_2 c + r$ for some $0 < r \leq c$, $q_1, q_2 \in \mathbb{N}_0$. Assume without loss of generality that $x_1 > x_2$. Thus $q_1 - q_2 \geq 1$, and therefore one can c-cut H_{x_1} to the option $O_{x_1} = (x_2, q_1 - q_2, \ldots, q_1 - q_2)$. Since c is even, by the definition of nim-sum, $\mathcal{G}(O_{x_1}) = \mathcal{G}(x_2)$, and thus, by definition of the mex-function, $\mathcal{G}(x_1) \neq \mathcal{G}(x_2)$. $\square$

In Section 2, we consider several families of partition games (e.g. those where $1 \notin \mathcal{C}$, or those with only odd values in $\mathcal{C}$) and prove their pure periodicity or pure arithmetic-periodicity. For the remaining families, many games seem to have a purely arithmetic-periodic behavior. To deal with them, we provide, in Section 3, a set of testing conditions whether a game is purely arithmetic-periodic, and apply them to particular instances. In Section 4, we give some background on classical games of TAKE-AND-BREAK, such as OCTAL and HEXADECIMAL games. Finally, in Section 5, we mention some remaining rulesets $\mathcal{C}$ for which a possible regularity of the $\mathcal{G}$-sequence remains open, as well as other open questions.

2 Particular partition games

In this section, we study some specific families of partition games with arithmetic-periodic Sprague-Grundy sequences. The notation $(m_1, \ldots, m_p)$ $(+s)$ describes the arithmetic-periodic sequence of *period* p and *saltus* s for which the first p values are $m_1, \ldots, m_p$. We write $(m_i, \ldots, m_j)^k$, if a subsequence $(m_i, \ldots, m_j)$ is repeated k times. Thus, for example, $(0, 1, 2)^2$ $(+3)$ denotes the arithmetic-periodic sequence of period 6 and saltus 3, *i.e.* $0, 1, 2, 0, 1, 2, 3, 4, 5, 3, 4, 5, \ldots$

As in Theorem 2, we will often encounter a unique decomposition of an integer via the division algorithm. For a given divisor (or period) $p \in \mathbb{N}$, for any $n \in \mathbb{N}$, there exists a unique ordered pair of nonnegative integers $(q_n, r_n) = (q, r)$ such that $0 < r \leq p$ and $n = pq + r$.[6] For consistency we will always use the letters q and r with this meaning, sometimes, but not necessarily, indexed to indicate their origin.

We first show that if $\mathcal{C}$ contains only odd numbers, including the possibility to split a heap into exactly two heaps, then the $\mathcal{G}$-sequence of $G(\mathcal{C})$ is purely periodic with period 2.

Proposition 3. *Let $\mathcal{C}$ be a (possibly infinite) ruleset consisting exclusively of odd cut-numbers, with $\min \mathcal{C} = 1$. Then $G(\mathcal{C}) = 0, 1, 0, 1, \ldots$, i.e. $\mathcal{G}(n) = 0$ if n is odd and $\mathcal{G}(n) = 1$ if n is even.*

Proof. Note that $\mathcal{G}(1) = 0$ and $\mathcal{G}(2) = 1$, since $\min \mathcal{C} = 1$.

Suppose that n is odd, and study a generic option of H_n. The number of heaps in O_n of odd size must be odd. Therefore the number of heaps of even size is also odd. By induction, this shows that $\mathcal{G}(O_n) = 1$.

Suppose next that n is even. The number of heaps of odd size in O_n must be even, and so the number of heaps of even size must also be even. By induction, this gives the claim for even heap sizes. $\square$

In this section, and later, we get repeated use of a very simple lemma.

---

[5]Such games allow at must one cut, but it might be combined with various removals; see Section 4 on class of TAKE-AND-BREAK.

[6]We use the convention $0 < r \leq p$, since H_1 is terminal in CUT.

Lemma 4. *Consider a ruleset $\mathcal{C}$ and a disjunctive sum $h = (k, h_1, \ldots, h_c)$. If all non-zero $\mathcal{G}$-values can be paired, except for $\mathcal{G}(k)$, then $\mathcal{G}(h) = \mathcal{G}(k)$. Moreover, if h is a c-cut of H_n, with $c \in \mathcal{C}$, then H_n has an option with $\mathcal{G}$-value $\mathcal{G}(k)$.*

Proof. Obvious. □

This result is useful to prove lower bounds on $\mathcal{G}$-values from given positions. (Especially since the proofs will tend to divide into several cases with similar arguments.) The upper bounds can be proved directly (using Lemma 1), or one can show that the anticipated value cannot have appeared before for smaller heap sizes. In either way, we rely heavily on induction.

In fact, for the upper bounds, an extension of Lemma 1 will prove useful. It reveals a general property of a combination of the division algorithm and the nim-sum.

Lemma 5. *Let $h = (h_0, \ldots, h_c)$ be a c-cut of $n = qp + r$,[7] and where, for all i, $h_i = pq_i + r_i$. If $c \geq p$, then $\bigoplus_{i=0}^{m} q_i \neq q$.*

Proof. By way of contradiction, suppose that $q = q_0 \oplus \cdots \oplus q_m$.

Since $\sum h_i = n$, we have

$$\sum_{i=0}^{c} (q_i c + r_i) = qc + r.$$

Note that $r \equiv \sum_{i=0}^{c} r_i \pmod{c}$ forces $r \leq \sum_{i=0}^{c} r_i$, and hence $\sum_{i=0}^{c} q_i \leq q$.

By combining this with the assumption and Lemma 1, we get $q = \bigoplus_{i=0}^{c} q_i \leq \sum_{i=0}^{c} q_i \leq q$, and hence

$$\bigoplus_{i=0}^{c} q_i = \sum_{i=0}^{c} q_i. \tag{2}$$

But then $c \geq p$ gives a contradiction. Namely, by (2) and for all i, $r_i > 0$, we get $r = \sum_{i=0}^{c} r_i > c \geq p$, which contradicts the definition of $r \leq p$. □

First, we study partition games, where a heap splits into at least three heaps. In this case, optimal play is reduced to using only $2 \leq c_1 = \min \mathcal{C}$, and the Sprague-Grundy sequence is purely arithmetic-periodic with period c_1 and saltus 1. Here, a cut-set may be infinite. For example, if $c_1 = 2$, we prove that $G(\mathcal{C}) = 0, 0, 1, 1, \ldots$ Let r be the smallest positive integer congruent with $n \pmod{c_1}$. We show that for all heaps H_n, $\mathcal{G}(n) = \dfrac{n-r}{c_1} = \left\lfloor \dfrac{n-1}{c_1} \right\rfloor$.

Proposition 6. *Consider a (possibly infinite) ruleset $\mathcal{C}$ with $c_1 = \min \mathcal{C} \geq 2$. Then, $G(\mathcal{C}) = (0)^{c_1} (+1)$.*

Proof. We want to prove that for every positive integer n, $\mathcal{G}(n) = q$, where $n = c_1 q + r$. Clearly $\mathcal{G}(1) = \mathcal{G}(2) = 0$. Assume the statement holds for all $n' < n$. By $c_1 = \min \mathcal{C}$, Lemma 5 gives that there is no option O_n with $\mathcal{G}(O_n) = q$.

It remains to prove that H_n has an option of $\mathcal{G}$-value g for all $g \leq q - 1$. There are two cases:

1. If c_1 is even, then, for each integer $g \leq q - 1$, let $O_n = (gc_1 + r, q - g, \ldots, q - g)$ be a c_1-cut (with c_1 copies of H_{q-g}). Here Lemma 4 applies, but we give the details since it is the first occurrence of the lemma. This is an option of H_n, since $gc_1 + r + (q-g)c_1 = qc_1 + r = n$. Furthermore, $\mathcal{G}(O_n) = \mathcal{G}(gc_1 + r) \oplus (c_1 \otimes \mathcal{G}(q-g)) = g \oplus 0 = g$, since by induction $\mathcal{G}(gc_1 + r) = g$ and by c_1 even.

2. If c_1 is odd, for each $\mathcal{G}$-value $g \leq q - 1$, we define a c_1-cut O_n, such that $\mathcal{G}(O_n) = g$. We have two subcases:

---

[7] That is, h is a $(c+1)$-partition of n.

2.1 If $q - g - 1$ is even, let

$$h_0 = gc_1 + r$$

$$h_j = \frac{1}{2}(q - g - 1)c_1 + 1 \qquad \text{for } j = 1, 2$$

$$h_j = 1 \qquad \text{for } 3 \leq j \leq c_1$$

Then $O_n = (h_0, \ldots, h_{c_1})$ is an option of H_n since $h_1 = h_2$ are non-negative integers, since $q - g$ is odd. Apply Lemma 4, using $\mathcal{G}(gc_1 + r) = g$ by induction and $\mathcal{G}(1) = 0$.

2.2 If $q - g - 1$ is odd, let

$$h_0 = gc_1 + r$$

$$h_j = \frac{1}{2}((q - g - 1)c_1 + 1) \qquad \text{for } j = 1, 2$$

$$h_j = 2 \qquad \text{for } j = 3$$

$$h_j = 1 \qquad \text{for } 4 \leq j \leq c_1$$

Apply Lemma 4 on $O_n = (h_0, \ldots, h_{c_1})$, using $\mathcal{G}(1) = \mathcal{G}(2) = 0$, and that, by induction $\mathcal{G}(gc_1 + r) = g$. (If $c_1 = 3$, then omit the last item.)

This proves that there is an option with $\mathcal{G}$-value g for all $0 \leq g < q$, and thus $\mathcal{G}(n) \geq q$. Together with (2), we obtain, for all n, $\mathcal{G}(n) = q$. $\qquad \square$

Next, we study partition games for which the players can split a heap into two, three or four heaps. In this case, even if the players are allowed to split a heap into more than four heaps, then the $\mathcal{G}$-sequence is purely arithmetic-periodic with period 1 and saltus 1.

Proposition 7. *Let $\mathcal{C}$ be a (possibly infinite) ruleset with $\{1, 2, 3\} \subseteq \mathcal{C}$. Then, for all heaps H_n, $\mathcal{G}(n) = n - 1$.*

Proof. For base cases, $\mathcal{G}(1) = 0$ and $\mathcal{G}(2) = 1$.

Apply Lemma 5 with $p = 1$, $r = 1$, $n - 1 = q$. Hence there is no option of n of $\mathcal{G}$-value $n - 1$.

To prove that $\mathcal{G}(n) \geq n - 1$, we must find an option O_n with $\mathcal{G}(O_n) = g$, for all $g < n - 1$. The 1-cut $O_n = (1, n - 1)$, satisfies $\mathcal{G}(O_n) = n - 2$ by induction. Otherwise, let $g < n - 2$, and apply Lemma 4. There are four cases depending on the parities of n and $g \leq n - 3$. If n and g have the same parity, we use a 3-cut, and otherwise a 2-cut. In all cases, we may use that, by induction, $\mathcal{G}(O_n) = \mathcal{G}(g + 1) = g$.

O_n	n odd	n even
g odd	$\left(g + 1, 1, \dfrac{n - g - 2}{2}, \dfrac{n - g - 2}{2}\right)$	$\left(g + 1, \dfrac{n - g - 1}{2}, \dfrac{n - g - 1}{2}\right)$
g even	$\left(g + 1, \dfrac{n - g - 1}{2}, \dfrac{n - g - 1}{2}\right)$	$\left(g + 1, 1, \dfrac{n - g - 2}{2}, \dfrac{n - g - 2}{2}\right)$

Altogether, $\mathcal{G}(n) = \mathrm{mex}(\{0, \ldots, n - 2\}) = n - 1$. $\qquad \square$

Finally, we study finite partition games, namely the CUT class where a player can split a heap into 2, 4 or $2k + 1$ heaps, where $k \geq 1$ is a given game parameter (note that this includes the game $G(1, 2, 3)$). In this case, the Sprague-Grundy sequence is purely arithmetic-periodic with period $2k$ and saltus 2. For example, if $k = 2$, then $G(1, 3, 4) = 0, 1, 0, 1, 2, 3, 2, 3, \ldots$ In general, we prove that $\mathcal{G}(n) = 2 \left\lfloor \dfrac{n}{2k} \right\rfloor + 1 - (n \bmod 2)$.

Proposition 8. *For any given $k \in \mathbb{N}$, let $\mathcal{C} = \{1, 3, 2k\}$. Then, $G(\mathcal{C}) = (0, 1)^k \; (+2)$.*

Proof. Given $k \in \mathbb{N}$, we must show that the period is $p = 2k$. Hence, for any heap H_n, define integers q, r by $n = 2kq + r$, $0 < r \leq 2k$.

More precisely, we prove that, for all heaps H_n, $\mathcal{G}(n) = 2q + 1 - (r \bmod 2)$.

Note that (for any k), $\mathcal{G}(1) = 0$ ($q = 0$ and $r = 1$) and $\mathcal{G}(2) = 1$ ($q = 0, r = 2$). Consider $n \geq 3$.

We show first that $\mathcal{G}(n) \leq 2q + 1 - (r \bmod 2)$.

If $\mathcal{G}(n) > 2q + 1 - r \pmod 2$, then H_n must have an option O_n with $c \in \mathcal{C}$ such that $\mathcal{G}(O_n) = 2q + 1 - (r \bmod 2)$.

In case $c = 2k$, assume that there is a c-cut $O_n = (n_{q_0}, \ldots, n_{q_c})$, with

$$2kq + r = n = \sum_{i=0}^{c}(2kq_i + r_i) = 2k\sum_{i=0}^{c} q_i + \sum_{i=0}^{c} r_i.$$

Then Lemma 5 applies with $p = 2k$, so $q \neq q_0 \oplus \cdots \oplus q_c$. Now,

$$2q + (1 - r \bmod 2) = \mathcal{G}(O_n) \tag{3}$$

$$= \bigoplus_{i=0}^{c} \mathcal{G}(n_{q_i}) \tag{4}$$

$$= \bigoplus_{i=0}^{c}(2q_i + (1 - r_i \bmod 2)) \tag{5}$$

$$= 2\bigoplus_{i=0}^{c} q_i + \bigoplus_{i=0}^{c}(1 - r_i \bmod 2), \tag{6}$$

The equality (4) holds by induction and (6) holds since 2 is a power of 2, and since, for all i, $(r_i \bmod 2) < 2$. By combining $q \neq q_0 \oplus \cdots \oplus q_c$ with $r \bmod 2 = \bigoplus_{i=0}^{c} r_i \bmod 2$, this gives a contradiction.

Therefore assume $c \in \{1, 3\}$. Then, by (6)

$$1 - r \bmod 2 = \bigoplus_{i=0}^{c}(1 - r_i \bmod 2) \tag{7}$$

$$= \left(\bigoplus_{i=0}^{c} r_i \bmod 2\right) \tag{8}$$

$$= r \bmod 2 \tag{9}$$

Equality (8) holds since c is odd, and (9) holds by $\sum_{i=0}^{c} r_i = r$, a contradiction.

Thus, there is no option of H_n with $\mathcal{G}$-value $2q + (1 - r \bmod 2)$.

We now prove that, from a heap H_n of $n = 2kq + r$ counters, there is an option of any $\mathcal{G}$-value smaller than $2q + (1 - r \bmod 2)$. Lemma 4 applies, but we give full details since the desired options vary a bit depending on various parities. There are two cases, depending on the parity of r.

1. If r is odd, then $2q + (1 - r \bmod 2) = 2q$, and

   1.1 for each integer $x \in [0, q - 1]$,

   $$O_n = (2kx + r, q - x, \ldots, q - x)$$

   is an option of H_n, obtained by a $2k$-cut. By induction, $\mathcal{G}(O_n) = 2x$, which gives the even $\mathcal{G}$-values in $[0, 2q - 2]$.

   1.2 if $r = 1$, for each integer $x \in [1, q - 1]$,

   $$O_n = (2kx, 1, (q - x)k, (q - x)k)$$

   is an option of H_n, obtained by a 3-cut. By induction, $\mathcal{G}(O_n) = 2x - 1$, which gives the odd $\mathcal{G}$-values in $[1, 2q - 3]$, and the value $2q - 1$ is obtained by the option $O_n = (2kq, 1)$.

1.3 if $r \geq 3$, for each integer $x \in [0, q-1]$,

$$O_n = (2kx + r - 1, 1, (q-x)k, (q-x)k)$$

is an option of H_n, obtained by a 3-cut. By induction, $\mathcal{G}(O_n) = 2x + (r \bmod 2) = 2x + 1$ since r is odd, which gives the odd $\mathcal{G}$-values in $[1, 2q-1]$.

Altogether, this implies $\mathcal{G}(n) \geq 2q$, if r is odd.

2. If r is even, then $2q + (1 - r \bmod 2) = 2q + 1$, and

2.1 for each integer $x \in [0, q-1]$,

$$O_n = (2kx + r + 1, q - x, \ldots, q - x)$$

is an option of H_n, obtained by a $2k$-cut. By induction, $\mathcal{G}(O_n) = 2x + 1$, which gives the odd $\mathcal{G}$-values in $[1, 2q-1]$.

2.2 for each integer $x \in [0, q-1]$,

$$O_n = (2kx + r, 1, (q-x)k, (q-x)k)$$

is an option of H_n obtained by a 3-cut. By induction, $\mathcal{G}(O_n) = 2x$, which gives the even $\mathcal{G}$-values in $[0, 2q-2]$, and the value $2q$ is obtained by the option $O_n = (2kq + r, 1)$.

Thus $\mathcal{G}(n) \geq 2q + 1$, for even r.

Hence, for all heaps H_n, $\mathcal{G}(n) = 2q + 1 - (r \bmod 2)$. $\qquad\square$

Note that when $k = 1$, Proposition 8 gives the same result as Proposition 7 when $k = 3$ (and as such, $\mathcal{C} = \{1, 2, 3\}$).

The above results cover a large range of partition games, but in remaining cases we were not able to have direct proofs. Yet, many of them seem to be well-behaved. The next section is devoted to building a test that allows us to prove (using a small number of computations) that a given game is purely arithmetic-periodic. We then use this test to prove that some games have a purely arithmetic-periodic Sprague-Grundy sequence.

3   An arithmetic-periodicity test for partition games

The purpose of this section is to provide an explicit tool to verify if a partitioning game is purely arithmetic-periodic by computing a small number of initial $\mathcal{G}$-values. Similar results are known as the SUBTRACTION, OCTAL and HEXADECIMAL periodicity tests (see Section 4.2, Theorem 16 and [13]), in the latter case concerning pure arithmetic-periodicity. Recall that for OCTAL, the number of computations to prove the periodicity is in the range of twice the period, whilst it takes at least 7 times the period to prove the arithmetic-periodicity of HEXADECIMAL (together with a couple of additional tests). For (finite) SUBTRACTION, the range of computations is given by the sum of the period and the highest number in the subtraction set. We review this development in Section 4.

In Section 3.1, we prove that computing at most the first $4p$ values of the $\mathcal{G}$-sequence (where p is the expected period, which should be determined by a blind computation) is enough to prove pure arithmetic-periodicity. We will also show that in some cases (depending on $\mathcal{C}$), the first $3p$ values are even sufficient (Section 3.2).

3.1   The AP-test

In this section, we describe a test that will be used to verify if a given partition game, on a finite ruleset, is purely arithmetic-periodic.

Definition 2 (Arithmetic-Periodic Test)**.** Consider a partition game on a finite ruleset $\mathcal{C}$, and a single heap H_n, with $n \in \mathbb{N}$. Suppose there exists a smallest positive integer p and a power of two $s = 2^t$, $t \in \mathbb{N}_0$, with $1 \leq s \leq p$, such that

AP1: if $n \leq 3p$ then $\mathcal{G}(n+p) = \mathcal{G}(n) + s$,

AP2: $\{\mathcal{G}(n) \mid n \leq p\} = \{0, 1, \ldots, s-1\}$, and

AP3: if $n \in [3p+1, 4p]$ then $\forall g \in [0, s-1] \; \exists \; c\text{-cut } O_n$ with $2 \leq c \in \mathcal{C}$ such that $\mathcal{G}(O_n) = g$.

Then the Arithmetic-Periodicity Test is (p,t)-verified for ruleset $\mathcal{C}$, and we write $\mathrm{AP}(\mathcal{C}) = (p,t)$.

The first two conditions are rather standard to prove the periodicity of TAKE-AND-BREAK games; similar conditions are required in the SUBTRACTION periodicity, OCTAL periodicity and HEXADECIMAL arithmetic-periodicity tests (see Section 4 for more discussion on this). However, contrary to those (when applicable), we require the saltus to be a power of two in order to prove arithmetic-periodicity. The third condition seems more unusual, and is used for the base case of the proof. We will see in the next subsection that, for some rulesets, the third condition AP3 can be directly deduced from AP1 and AP2, which suggests that there could be a periodicity test closer to the existing tests for subtraction, octal and hexadecimal games. We now state the main result of this section.

Theorem 9. *Suppose that $AP(\mathcal{C}) = (p,t)$, where $\max \mathcal{C} \leq 4p$. Then $G(\mathcal{C})$ is arithmetic-periodic with period p and saltus 2^t, i.e. for all heaps H_n, $\mathcal{G}(n+p) = \mathcal{G}(n) + 2^t$.*

In other words, if a partitioning game verifies the AP-test, then it is purely arithmetic-periodic. Note that both in the definition of the AP-test as well as in the main theorem, the saltus of the $\mathcal{G}$-sequence is a power of 2. The bound $\max \mathcal{C} \leq 4p$ ensures that AP3 applies as a base case for induction.

In order to prove this result, we use a couple of lemmas. As before, we will frequently make use of the fact that for every heap H_n, and a given period p, there exists a unique pair $(q, r) = (q_n, r_n)$ such that $n = pq + r$ with $q \in \mathbb{N}_0$ and $0 < r \leq p$. The next result gives the closed formula corresponding to a purely arithmetic-periodic $\mathcal{G}$-sequence. (It applies to CUT, but also any other ruleset.)

Lemma 10. *Consider any heap game. Suppose there exists a period p, a saltus s, with, for all $n > p$, $\mathcal{G}(n) = \mathcal{G}(n-p) + s$. Then, for all $n > 0$, $\mathcal{G}(n) = sq_n + \mathcal{G}(r_n)$.*

Proof. For all $1 \leq n \leq p$, $n = r$ with $q = 0$, and hence, as a base case, $\mathcal{G}(n) = \mathcal{G}(r)$.

Let $n = pq + r > p$. The $\mathcal{G}$-value of H_n is

$$
\begin{aligned}
\mathcal{G}(n) &= \mathcal{G}(n-p) + s \\
&= \mathcal{G}(p(q-1) + r) + s \\
&= s(q-1) + \mathcal{G}(r) + s \\
&= sq + \mathcal{G}(r),
\end{aligned}
\tag{10}
$$

where (10) is by induction. $\qquad\square$

One nice consequence of Lemma 10 is that the $\mathcal{G}$-values decompose, provided that the saltus is a power of 2, and this simple result outlines our approach.

Lemma 11. *Consider a ruleset $\mathcal{C}$ and a disjunctive sum of heaps $H = (h_0, \ldots, h_c)$. Suppose that Lemma 10 is satisfied, with $h_i \leq n$, for all $0 \leq i \leq c$, and with the two additional constraints*

- $s = 2^t$, $t \in \mathbb{N}_0$,

- $\mathcal{G}(n) < s$ for all $1 \leq n \leq p$.

Then

$$
\mathcal{G}(H) = (q_0 \oplus \cdots \oplus q_c)s + \mathcal{G}(r_0) \oplus \cdots \oplus \mathcal{G}(r_c),
\tag{11}
$$

where for all i, $(q_i, r_i) = (q_{h_i}, r_{h_i})$.

Proof.

$$
\begin{aligned}
\mathcal{G}(H) &= \mathcal{G}(q_0 p + r_0, \ldots, q_c p + r_c) \\
&= \mathcal{G}(q_0 p + r_0) \oplus \cdots \oplus \mathcal{G}(q_c p + r_c) \\
&= (q_0 s + \mathcal{G}(r_0)) \oplus \cdots \oplus (q_c s + \mathcal{G}(r_0)) \\
&= (q_0 \oplus \cdots \oplus q_c)s + \mathcal{G}(r_0) \oplus \cdots \oplus \mathcal{G}(r_c),
\end{aligned}
\tag{12}
\tag{13}
\tag{14}
\tag{15}
$$

where (13) is by Sprague-Grundy theory, (14) is by Lemma 10, and (15) is by the two constraints. $\quad\square$

Lemma 11 becomes useful in proving that the AP-test implies arithmetic-periodicity by providing invariance of $\mathcal{G}$-values along arithmetic progressions of heap sizes. Namely the remainder part of the heap sizes in specified options can remain the same, whereas selected parts of the dividend part can be cancelled out, in a similar sense as in Lemma 4. This observation will appear a couple of times in the proof of Theorem 9. This theorem will be proved by induction, with a rather technical base case, which will make use of the condition $c_k \geq 2$ in AP3. We consider a part of this base case in the following lemma.

Lemma 12. *Let $\mathcal{C} = \{c_1, \dots, c_k\}$ be a ruleset with $c_k \geq 2$ such that $AP(\mathcal{C}) = (p, t)$, with $s = 2^t$, $t \in \mathbb{N}_0$. Then for each H_n with*

1. *$n \in [3p+1, 4p]$ and for each $\mathcal{G}$-value $g \in [0, 2s-1]$, there is an option $O_n = (h_0, \dots, h_c)$, with $c \geq 2$ such that $\mathcal{G}(O_n) = g$,*

2. *$n \in [2p + 1, 3p]$ and for all $\mathcal{G}$-values $g \in [0, s-1]$, there is an option $O_n = (h_0, \dots, h_c)$, with $c \geq 2$ such that $\mathcal{G}(O_n) = g$.*

Proof. Case 1: Let H_n be such that $n = 3p + r_n \in [3p+1, 4p]$ and let $g \in [0, 2s-1]$. AP3 implies that for each $g \in [0, s-1]$, there is an option O_n such that $\mathcal{G}(O_n) = g$.

By the conditions AP1 and AP2, Lemma 10 gives that $\mathcal{G}(n) = 3s + \mathcal{G}(r_n)$ and hence, by the mex rule, for each $g \in [s, 2s-1]$, there is an option O_n such that $\mathcal{G}(O_n) = g$. Therefore, if $1 \notin \mathcal{C}$, there is nothing to prove. Consequently, it suffices to prove that if $1 \in \mathcal{C}$, and $O_n = (h_0, h_1)$ is an option of n obtained by a 1-cut, then $\mathcal{G}(O_n) \notin [s, 2s-1]$.

Assume $1 \in \mathcal{C}$ and consider the 1-cut $O_n = (h_0, h_1)$. There exist four unique nonnegative integers q_0, r_0, q_1, r_1 such that $0 < r_0, r_1 \leq p$ and $O_n = (q_0 p + r_0, q_1 p + r_1)$. As O_n is an option of n we have

$$(q_0 + q_1)p + r_0 + r_1 = n = 3p + r$$

which gives

$$r_0 + r_1 - r = (3 - q_0 - q_1)p.$$

As $0 \leq q_0 + q_1 \leq 3$ and $r_0 + r_1 \leq 2p$, we have on one hand $0 \leq r_0 + r_1 - r < 2p$ and on the other hand $r_0 + r_1 - b \equiv 0 \pmod{p}$. Hence $r_0 + r_1 - b \in \{0, p\}$. If it equals 0 then $q_0 + q_1 = 3$, and otherwise $q_0 + q_1 = 2$. Without loss of generality the possible values for q_0, q_1 and $q_0 \oplus q_1$ are summarized in the following table:

q_0	q_1	$q_0 \oplus q_1$
0	2	2
	3	3
1	1	0
	2	3

In particular, $q_0 \oplus q_1 \neq 1$. And, by Lemma 11, property (11) we have that

$$\mathcal{G}(O_n) = (q_0 \oplus q_1)s + \mathcal{G}(r_0) \oplus \mathcal{G}(r_1) \notin [s, 2s-1],$$

since s is a power of two and $\mathcal{G}(r_0), \mathcal{G}(r_1) < s$.

Case 2: Let $n \in [2p+1, 3p]$ and $g \in [0, s-1]$, and let $n' = n + p \in [3p+1, 4p]$ and $g' = g + s \in [s, 2s-1]$.

By the first part of the proof, we know that there is an option of $H_{n'}$,

$$O_{n'} = (q_{0,n'} p + r_{0,n'}, \dots, q_{m,n'} p + r_{m,n'})$$

such that $m \geq 2$ and $\mathcal{G}(O_{n'}) = g'$. Let $N = q_{0,n'} \oplus \cdots \oplus q_{m,n'}$, $R = \mathcal{G}(r_{0,n'}) \oplus \cdots \oplus \mathcal{G}(r_{m,n'})$ and $S = q_{0,n'} + \cdots + q_{m,n'}$. We apply Lemma 11 to $O_{n'}$ and get $\mathcal{G}(O_{n'}) = Ns + R$. Moreover $N = 1$, by $g' \in [s, 2s-1]$.

Define a disjunctive sum of heaps $h = (h_0, \dots, h_c)$ by

$$h_0 = r_{0,n'}$$

$$h_j = \frac{1}{2}(S-1)p + r_{j,n'} \qquad \text{for } j = 1, 2$$

$$h_j = r_{j,n'} \qquad\qquad \text{for } 3 \leq j \leq c$$

If $c = 2$, ignore the third part. Note that $S - 1 \geq 0$ is even by Lemma 1. The disjunctive sum h is an option of H_n since

$$
\begin{aligned}
h_0 + \cdots + h_c &= (S - 1)p + (r_{0,n'} + \cdots + r_{c,n'}) \\
&= n' - p \\
&= n
\end{aligned}
$$

Again, since $S - 1$ is even,

$$
\begin{aligned}
\mathcal{G}(h) &= R \\
&= \mathcal{G}(O_{n'}) - s \\
&= g' - s \\
&= g
\end{aligned}
$$

Hence, $h = O_n$ is indeed an option of n with $c \geq 2$ and $\mathcal{G}(O_n) = g$. $\qquad\square$

We can now prove Theorem 9: if a partition game verifies the AP-test, then its $\mathcal{G}$-sequence is purely arithmetic-periodic.

Proof of Theorem 9. Assume $\mathrm{AP}(\mathcal{C}) = (p, t)$, with $2^t = s$. Thus for $0 \leq q < 4$ where $n = qp + r \in [qp + 1, (q + 1)p]$, by Lemma 10, we have $\mathcal{G}(n) = \mathcal{G}(qp + r) = qs + \mathcal{G}(r)$.

We will now prove by induction that for any heap H_n, with $n = qp + r \geq 1$, $\mathrm{AP}(\mathcal{C}) = (p, t)$ implies the following two properties (where (A) implies the result, and where (B) is an auxiliary means for the induction).

(A) $\mathcal{G}(n) = qs + \mathcal{G}(r)$ and

(B) for all $\mathcal{G}$-values $g \in [0, (q - 1)s - 1]$, there is an option $O_n = (h_0, \ldots, h_c)$ such that $c \geq 2$ and $\mathcal{G}(O_n) = g$.

We have a base case. By Lemma 10, (A) holds for all $n \leq 4p$. Moreover, by Lemma 12, (B) holds for $q = 2, 3$, and it is trivially true for $q \leq 1$. So (B) holds for $n \leq 4p$.

For the induction step, let $n = qp + r > 4p$. We will prove the result in two steps: first we show that $\mathcal{G}(n) \geq qs + \mathcal{G}(r)$ together with (B), and then we show that $\mathcal{G}(n) \leq qs + \mathcal{G}(r)$.

1. Case $\mathcal{G}(n) \geq qs + \mathcal{G}(r)$: By induction, the heap of size $n' = n - 2p = q'p + r'$ verifies conditions (A) and (B). Consider any $\mathcal{G}$-value $g < (q' - 1)s$. Then, by (B), there is an option $O_{n'} = (q_{0,n'}p + r_{0,n'}, \ldots, q_{c,n'}p + r_{c,n'})$, with $c \geq 2$ and $\mathcal{G}(O_{n'}) = g$. Let $N = q_{0,n'} \oplus \cdots \oplus q_{m,n'}$, $S = q_{0,n'} + \cdots + q_{c,n'}$ and $R = \mathcal{G}(r_{0,n'}) \oplus \cdots \oplus \mathcal{G}(r_{c,n'})$. Define $O_n = (h_0, \ldots, h_c)$ by

$$
\begin{aligned}
h_0 &= Np + r_{0,n'} \\[4pt]
h_j &= \frac{1}{2}(S - N + 2)p + r_{j,n'} \qquad \text{for } j = 1, 2 \\[4pt]
h_j &= r_{m,n'} \qquad\qquad\qquad\qquad \text{for } 2 < j \leq c
\end{aligned}
$$

If $c = 2$, then omit the third part. This is an option of H_n with at least two cuts since $h_0 + \cdots + h_c = (2 + S)p + r_{0,n'} + \cdots + r_{c,n'}$ and its $\mathcal{G}$-value is $\mathcal{G}(O_n) = Ns + R = g$, by Lemma 11 (since the contributions $(S - N + 2)s/2$ for $j = 1, 2$ cancel out).

Hence H_n has options to all $\mathcal{G}$-values in $[0, (q' - 1)s - 1]$, i.e. $\mathcal{G}(n) \geq (q - 3)s$.

We now change O_n into $\hat{O}_n = (\hat{h}_0, \ldots, \hat{h}_c)$ as follows:

$$
\begin{aligned}
\hat{h}_0 &= q_{0,n'}p + r_{0,n'} + 2p \\[4pt]
\hat{h}_j &= q_{j,n'}p + r_{j,n'} \qquad\qquad \text{for } 0 < j \leq c
\end{aligned}
$$

11

This is an option of H_n with a c-cut since $\hat{h}_0 + \cdots + \hat{h}_m = n' + 2p$. Its $\mathcal{G}$-value is $\mathcal{G}(\hat{O}_n) = (N+2)s + R = g + 2s$.

Hence H_n has options of $\mathcal{G}$-values in $[2s, (q-1)s-1]$.

If $q > 4$ then, by putting together the two cases, H_n has options to all $\mathcal{G}$-values in $[0, (q-1)s-1]$, with $c \geq 2$ and (B) holds. Otherwise, if $q = 4$, then we take an option $O_{n'} = (q_{0,n'}p + r_{0,n'}, \ldots, q_{c,n'}p + r_{m,n'})$ of $n' = 3p + r = n - p$ with $\mathcal{G}$-value $g \in [0, s-1]$ and $c \geq 2$, which exists by Lemma 12. We denote by $S = q_{0,n'} + \cdots + q_{c,n'}$, $N = q_{0,n'} \oplus \cdots \oplus q_{c,n'}$ and $R = \mathcal{G}(r_{0,n'}) \oplus \cdots \oplus \mathcal{G}(r_{c,n'})$, and transform $O_{n'}$ into an option $O_n = (h_0, \ldots, h_c)$ by setting

$$h_0 = (N+1)p + r_{0,n'}$$

$$h_j = \frac{1}{2}(S-N)p + r_{j,n'} \qquad \text{for } j = 1, 2$$

$$h_j = r_{j,n'} \qquad\qquad \text{for } 3 \leq j \leq c$$

This is an option of H_n since $h_0 + \cdots + h_c = (S+1)p + r_{0,n'} + \cdots + r_{c,n'} = n' + p = n$ and its $\mathcal{G}$-value is $\mathcal{G}(O_n) = \mathcal{G}(O_{n'}) + s = g + s$.

This gives that for $q = 4$, H_n has options obtained by c-cuts, $c \geq 2$, to all $\mathcal{G}$-values in $[0, (q-1)s] = [0, 3s]$. Hence H_n verifies (B) in every case.

For the remaining part of the proof of (A), let $n' = n - (q-1)p = p + r$ and $g \in [0, s + \mathcal{G}(r) - 1]$.

Let $O_{n'} = (q_{0,n'}p + r_{0,n'}, \ldots, q_{c,n'}p + r_{c,n'})$ be an option of $H_{n'}$ such that $\mathcal{G}(O_{n'}) = g$. It exists, since $H_{n'}$ verifies (B) by induction. Note that as $n' \leq 2p$, if there is a j such that $q_{j,n'} \neq 0$, then it is unique. Hence, without loss of generality, assume that $q_{0,n'} \in \{0, 1\}$ and for $j > 0$, $q_{j,n'} = 0$. Therefore, if $R = \mathcal{G}(r_{0,n'}) \oplus \cdots \oplus \mathcal{G}(r_{m,n'})$, then $\mathcal{G}(O_{n'}) = q_{0,n'}s + R$ by Lemma 11.

Define $O_n = (h_0, \ldots, h_c)$ by

$$h_0 = (q_{0,n'} + q - 1)p + r_{0,n'}$$

$$h_j = r_{j,n''} \qquad\qquad \text{for } j > 0$$

This is an option of H_n, since $h_0 + \cdots + h_c = (q_{0,n'} + q - 1)p + r_{0,n'} + r_{1,n'} + \cdots + r_{c,n'} = n' + (q-1)p = n$. Its $\mathcal{G}$-value is $\mathcal{G}(O_n) = (q_{0,n'} + q - 1)s + R = g + (q-1)s$. Hence H_n has options to all $\mathcal{G}$-values in $[(q-1)s, qs + \mathcal{G}(r) - 1]$. Altogether, H_n has options to all $\mathcal{G}$-values in $[0, qs + \mathcal{G}(r) - 1]$.

2. Case $\mathcal{G}(n) \leq qs + \mathcal{G}(r)$:

Assume, as a contradiction, that $O_n = (q_0 p + r_0, \ldots, q_c p + r_c)$ is an option of H_n with $\mathcal{G}$-value $\mathcal{G}(O_n) = qs + \mathcal{G}(r)$, and where as usual $n = qp + r$. Let $N = q_0 \oplus \cdots \oplus q_c$ and $R = \mathcal{G}(r_0) \oplus \cdots \oplus \mathcal{G}(r_c)$. With this notation, $\mathcal{G}(O_n) = Ns + R$, by Lemma 11. Hence $q = N$ and

$$\mathcal{G}(r) = R, \tag{16}$$

since $s = 2^t$ and $\mathcal{G}(r_i), \mathcal{G}(r) < s$, by AP2.

Define $O_{n'} = (h_0', \ldots, h_c')$ by

$$h_0' = (q-2)p + r_0$$

$$h_j' = r_j, \qquad\qquad \text{for } j > 1.$$

This is an option of $n' = n - 2p$, and its $\mathcal{G}$-value is

$$\mathcal{G}(O_{n'}) = (q-2)s + R \tag{17}$$
$$= qs + \mathcal{G}(r) - 2s \tag{18}$$
$$= \mathcal{G}(n'), \tag{19}$$

where (17) and (19) are by Lemma 11, and (18) is by (16), a contradiction.

Putting together steps (1) and (2), the heap H_n verifies (A).

$\square$

3.2 Relaxed conditions on the AP-test

We now prove that for some families of games, the conditions AP1 and AP2 of the AP-test imply the condition AP3. We first prove that this is the case if the players are allowed to split a heap into at least one even number and one odd number of heaps. Again, Lemma 4 applies in the proof, but the situation gets a little technical, so we write out all details.

Proposition 13. *Let $\mathcal{C}$ be a ruleset with $|\mathcal{C}| \geq 2$. If $G(\mathcal{C})$ (p,t)-verifies AP1 and AP2, and there are cut numbers $c, c' \in \mathcal{C}$ of different parities with $2 \leq c, c' \leq 2p+1$, then $AP(\mathcal{C}) = (p,t)$.*

Proof. Let $s = 2^t$. It suffices to prove that $G(\mathcal{C})$ (p,t)-verifies AP3. We prove that for all heap sizes $n \in [3p+1, 4p]$ and all $\mathcal{G}$-values $g \in [0, s-1]$, there is an option $O_n = (h_0, \ldots, h_c)$ such that $c \geq 2$ and $\mathcal{G}(O_n) = g$. As usual, write $n = 3p + r$ with $0 < r \leq p$.

By AP2, for any $g \in [0, s-1]$, there exists a $k \in [1,p]$ such that $\mathcal{G}(k) = g$. Let $n' = n - k = 3p + r - k$. The desired cuts will depend on the parity of n'.

Case n' even: let $c \geq 2$ be even, and let (q', r') be the unique ordered pair such that $0 < r' \leq c$ and $n' = cq' + r'$. In particular, r' is even, since c and n' are also even. Moreover $q' > 0$ since $c \leq 2p < n'$. We define $h = (h_0, \ldots, h_c)$ by

$$h_0 = k$$

$$h_j = q + \frac{1}{2}r \qquad \text{for } j = 1, 2$$

$$h_j = q \qquad \text{for } 3 \leq j \leq c$$

(If $c = 2$, then the third part does not apply.) Then $h = O_n$ is an option of H_n since all $h_j \in \mathbb{N}$, and since $h_0 + \cdots + h_c = k + cq + r = k + n' = n$. Moreover, in the expression $\mathcal{G}(h_0) \oplus \cdots \oplus \mathcal{G}(h_c)$, the values $\mathcal{G}(h_1)$ and $\mathcal{G}(h_3)$ appear an even number of times, which gives directly $\mathcal{G}(O_n) = \mathcal{G}(k) = g$.

Case n' odd: let $c \geq 3$ be odd and let (q, r) be such that $0 \leq r < c$, $n' = cq' + r'$. Note that $q' > 0$ since $c \leq 2p < n'$. As n' and c are odd, either q' is even and r' is odd or vice versa.

- if q' is even and r' is odd, define the option $O_n = (h_0, \ldots, h_c)$ by:

$$h_0 = 1 + k$$

$$h_j = \frac{3}{2}q' + \frac{1}{2}(r' - 1) \qquad \text{for } j = 1, 2$$

$$h_j = 1 \qquad \text{for } j = 3$$

$$h_j = q' \qquad \text{for } 4 \leq j \leq c$$

If $c = 3$ then we only take the four first heaps. The option O_n is an option of n since $h_0 + \cdots + h_c = 1 + c + 3q' + r' - 1 + 1 + (c - 1 - 2)q' = 1 + k + q'_2 + r'_2 = 1 + c + n' = n$. In the expression $\mathcal{G}(h_0) \oplus \cdots \oplus \mathcal{G}(h_c)$ the terms $\mathcal{G}(h_1)$ and $\mathcal{G}(h_4)$ appear an even number of times and $\mathcal{G}(h_3) = 0$, and hence $\mathcal{G}(O_n) = 1 + k = g$.

- if q' is odd and r' is even, we define the option O_n by:

$$h_0 = 1 + k$$

$$h_j = \frac{1}{2}(3q' - 1) + \frac{1}{2}r' \qquad \text{for } j = 1, 2$$

$$h_j = 1 \qquad \text{for } j = 3$$

$$h_j = q' \qquad \text{for } 4 \leq j \leq c$$

it is an option of n since $h_0 + \cdots + h_c = 1 + k + 3q' - 1 + r + 1 + (c-3)q' = 1 + k + cq' + r' = n$. In the expression $\mathcal{G}(h_0) \oplus \cdots \oplus \mathcal{G}(h_c)$ the terms $\mathcal{G}(h_1)$ and $\mathcal{G}(h_4)$ appear an even number of times and $\mathcal{G}(h_3) = 0$. Hence $\mathcal{G}(O_n) = g$.

In each case, there is an option O_n of H_n obtained by a c-cut, $c \geq 2$, such that $\mathcal{G}(O_n) = g$, *i.e.* $G(\mathcal{C})$ (p, t)-verifies AP3, which means that $AP(\mathcal{C}) = (p, t)$. $\square$

If $|\mathcal{C}| = 2$ and one possibility is to split a heap into exactly two heaps, and the other is to split into a a given odd number of heaps, then sometimes AP1 and AP2 imply AP3.

Proposition 14. *Let $\mathcal{C} = \{1, c\}$ with $c \geq 4$ even, and suppose that $G(\mathcal{C})$ (p, t)-verifies AP1 and AP2, with $c \leq p$. If there exist heap sizes $1 \leq x_1, x_2 \leq p/2$ of different parity, such that $\mathcal{G}(x_1) = \mathcal{G}(x_2)$, then $AP(\mathcal{C}) = (p, t)$.*

Proof. We prove that the game $G(\mathcal{C})$ (p, t)-verifies AP3, *i.e.*, for any heap H_n with $n \in [3p+1, 4p]$ and for any $\mathcal{G}$-value $g \in [0, 2^t - 1]$, there exists an option O_n, such that $\mathcal{G}(O_n) = g$. Since AP2 is (p, t)-verified, this can be done by proving that, for all heap sizes $k \in [1, p]$, there exists an option O_n such that $\mathcal{G}(O_n) = \mathcal{G}(k)$.

The proof is divided into four cases depending on the parities of k and n, and we summarize the relevant c-cuts in a table:

O_n	n odd	n even
k odd	$(k, f(n,k), f(n,k), 1, \ldots, 1)$	$(k, x_1, x_2, f(n,k), f(n,k), 1, \ldots, 1)$
k even	$(k, x_1, x_2, f(n,k), f(n,k), 1, \ldots, 1)$	$(k, f(n,k), f(n,k), 1, \ldots, 1)$

Here

$$f(n, k) = \frac{n - k - c}{2} + 1$$

if n and k have the same parity, and

$$f(n, k) = \frac{n - k - c - x_1 - x_2 + 5}{2}$$

if n and k have different parities and $c \geq 6$. If $c = 4$, then by say $0 < x_1 < x_2$ different parities, we consider instead the option $O_n = (k, x_1, x_2 - 1, f(n,k), f(n,k))$. It is straightforward to justify in each case that O_n is an option, and then apply Lemma 4.

Hence, for all heap sizes $k \in [1, p]$, there exists an option of n with the same $\mathcal{G}$-value. This implies that the condition AP3 is (p, t)-verified, and thus that the AP-test is (p, t)-verified for $G(\mathcal{C})$. $\square$

We now prove that the conditions of Proposition 14 are verified for all those games as long as the period is lower bounded by $4c + 3 \leq p$.

Corollary 15. *Let $\mathcal{C} = \{1, c\}$ with $c \geq 4$ even. If $G(\mathcal{C})$ (p, t)-verifies AP1 and AP2 for some $p \geq 4c+3$, then $AP(\mathcal{C}) = (p, t)$.*

Proof. By Proposition 14, it suffices to prove that there exists $x_1, x_2 < p/2$ of different parities, such that $\mathcal{G}(x_1) = \mathcal{G}(x_2)$.

Note that $\mathcal{G}(2) = 1$, since the only option is $(1,1)$ which has $\mathcal{G}$-value 0. Hence we can assume $x_2 = 2$.

We claim that we can choose $x_1 = 2c + 1$ to get $\mathcal{G}(x_1) = \mathcal{G}(2) = 1$. In order to do that, we prove that the beginning of the $\mathcal{G}$-sequence of the game $G(\mathcal{C})$ is $(0,1)^{c/2}$ and the following c values are different from 1 and 0, and the $2c + 1$-th value is 1. If $k \leq c$, then $\mathcal{G}(k) = (k - 1) \pmod 2$, by Proposition 3.

Now, let $k \in [c+1, 2c]$. If k is odd, then k admits the 1-cut option $(k - c, c)$ of Grundy value 1 since c is even, and the c-cut option $(k - c, 1, \ldots, 1)$ of Grundy value 0. If k is even, it admits the c-cut option $(k - c, 1, \ldots, 1)$ of Grundy value 1, and the 1-cut option $(k/2, k/2)$ of Grundy value 0. It thus implies that $\mathcal{G}(k) > 1$.

Finally, we prove $\mathcal{G}(2c + 1) = 1$. We now set $k = 2c + 1$.

From k, one can reach the value 0 by the option $(1, 2, \ldots, 2)$ obtained by a c-cut. All the 1-cuts (i_0, i_1) are such that without loss of generality $i_0 > c$ and $i_1 \leq c$, so $\mathcal{G}(i_0, i_1) \neq 1$ since $\mathcal{G}(i_1) < 2$ and $\mathcal{G}(i_2) \geq 2$.

It suffices to prove that there is no c-cut $O_k = (i_0, \ldots, i_c)$ with $\mathcal{G}(O_k) = 1$. If there is some j such that $i_j > c$, then it is unique, and as before $\mathcal{G}(i_j) \geq 2$ implies $\mathcal{G}(O_k) \geq 2$. Hence, assume, for all j, $i_j \leq c$. If the $\mathcal{G}$-value of the c-cut were 1, then we must have an odd number of even heap sizes, and hence an even number of odd heap sizes, since c is even. But $2c + 1$ is odd, so the c-cut cannot sum up correctly. Therefore, $\mathcal{G}(2c + 1) = 1$. Moreover, $2c + 1 < p/2$ since $4c + 3 \leq p$. $\qquad \square$

3.3 Applications of the AP-test

Table 1 summarizes applications of the AP-test for some partitioning games (the games already solved in Section 2 are not in the table). All the games in this list satisfy the test and hence are proved to be purely arithmetic-periodic. More specifically, Corollary 15 has been applied to the games $\{1, 4\}$, $\{1, 6\}$, $\{1, 8\}$, and $\{1, 10\}$. We note that for games of the form $\{1, c\}$, there seem to be rather long periods depending on c, with always the same saltus. We wonder whether this regularity holds for higher values of c.

Open Problem 1. Given $c \geq 2$, the game $G(\mathcal{C})$ with $\mathcal{C} = \{1, 2c\}$ is arithmetic-periodic of length $12c$ and saltus 8.

Surprisingly, when one adjoins new values to the games $\{1, 2c\}$ (with $c \geq 2$), the period and saltus may change significantly. This is for example the case for the game $\{1, 4\}$, which has period 24 and saltus 8, but $\{1, 3, 4\}$ has period 4 and saltus 2, while $\{1, 4, 9\}$ has period 40 and saltus 16.

The cases where $1, 2 \in \mathcal{C}$ but $3 \notin \mathcal{C}$ remain the hardest to understand. If Table 1 suggests a purely arithmetic-periodic behavior when $|\mathcal{C}| \geq 3$, we did not detect any general pattern, depending on $\mathcal{C}$. For example, when $|\mathcal{C}| = 3$, the games $\{1, 2, 4\}$ and $\{1, 2, 6\}$ have identical Sprague-Grundy sequences, whereas $\{1, 2, 5\}$ and $\{1, 2, 7\}$ are more singular. Even worse, the games $\{1, 2, 8\}$ and $\{1, 2, 7, 8\}$ seem to be ultimately arithmetic-periodic with saltus 8 and a preperiod of positive length equal to 6 (which is not the case of the other sequences we computed).[8] A proof of this result remains to be done, as the AP-test does not cover ultimate behavior (and does not seem to be easily adapted for them).

4 A review of TAKE-AND-BREAK in relation with CUT

We review some history of theory intertwining SUBTRACTION with CUT, and such games are gathered under the umbrella TAKE-AND-BREAK. (Then in Section 5, we return to conclude our findings in this broader perspective.)

4.1 TAKE-AND-BREAK

TAKE-AND-BREAK [3] is a large class of 2-player impartial combinatorial games with alternating play. A game position is represented by a finite multiset of heaps. A move consists in choosing a single heap, removing some beans, and possibly splitting the remaining heap into several heaps. If splitting is not allowed, we have (pure) SUBTRACTION. In this case, the rules are given by a set $\mathcal{S}$ of positive integers

---

[8]These are the only examples we found with a non-0 preperiod.

CUT-set	Sprague-Grundy sequence
$\{1,4\} \cup K$ with $K \subseteq \{6,8,10\}$	$((0,1)^2(2,3)^2,1,4,5,4,(3,2)^2(4,5)^2(6,7)^2)\ (+8)$
$\{1,6\} \cup K$ with $K \subseteq \{8,10\}$	$((0,1)^3(2,3)^3,1,4,(5,4)^2(3,2)^3(4,5)^3(6,7)^3)\ (+8)$
$\{1,8\}$	$((0,1)^4(2,3)^4,1,4,(5,4)^3(3,2)^4(4,5)^4(6,7)^4)\ (+8)$
$\{1,10\}$	$((0,1)^5(2,3)^5,1,4,(5,4)^4(3,2)^5(4,5)^5(6,7)^5)\ (+8)$
$\{1,4\} \cup K$ with $K \subseteq \{3,5,6,7,8\}$ and 3, 5 or $7 \in K$	$(0,1)^2\ (+2)$
$\{1,6\} \cup K$ with $K \subseteq \{3,5,7,8\}$ and 3, 5 or $7 \in K$	$(0,1)^3\ (+2)$
$\{1,8\} \cup K$ with $K \subseteq \{3,5,7\}, K \neq \emptyset$	$(0,1)^4\ (+2)$
$\{1,2,4\} \cup K, \{1,2,6\} \cup K'$ with $K \subseteq \{6,7,8\}, K' \subseteq \{7,8\}$	$(0,1,2,3,1,4,3,2,4,5,6,7)\ (+8)$
$\{1,2,5\} \cup K$ with $K \subseteq \{4,6,7,8\}$	$(0,1,2,3,1,4,3,6,4,5,6,7)\ (+8)$
$\{1,2,7\}$	$(0,1,2,3,1,4,3,2,4,5,6,7,8,9,7,6,9,8,11,10,12,13,10,11,13,12,15,14)\ (+16)$
$\{1,4,9\}$	$(0,1,0,1,2,3,2,3,1,4,5,4,3,6,7,6,4,5,8,9,6,7,10,11,9,$ $8,9,12,11,10,11,14,12,13,12,13,14,15,14,15)\ (+16)$

Table 1: Some partition games for which the purely arithmetic-periodicity is proved with the AP-test.

which specifies the number of tokens that can be removed from the heap. When the heap may be split, the rulesets are often given by a code that specifies how many tokens can be removed and the number of heaps that the remainder of the heap can be split into.

Let us review some of the history of TAKE-AND-BREAK. The origin to the line of research was the invention of a fairy game of CHESS in the 1930s, called DAWSON'S CHESS [5], named by its inventor, played by using only the pawns of CHESS and with mandatory captures. It was soon remarked that the game is equivalent to a game on heaps of counters, where a player may remove (i) a single counter if by removing it no heap remains, (ii) two counters if no heap or a single heap remain (iii) three counters if no heap, one heap or two heaps remain. In the normal play convention it was discovered that the solution has a regular behavior (ultimately periodic) [11]. The elegance of the solution of DAWSON'S CHESS, together with the similar game of KAYLES [11] (remove one or two counters anywhere), gave rise to classifications of games on heaps of counters [11].

For example, the family of games for which a heap can be split into at most two heaps is nowadays called OCTAL games. This name is due to an explicit way to express any ruleset with an octal code $\mathbf{d_0}.\mathbf{d_1} \cdots \mathbf{d_k}$ with d_i an integer, $0 \leq d_i \leq 7$ for $1 \leq i \leq k$. More precisely, each value d_i with $i > 0$ is encoded in binary as $d_i = 2^0 e_0 + 2^1 e_1 + 2^2 e_2$, where each $e_j \in \{0,1\}$. The ruleset allows to remove i tokens from a heap and split the remainder into j non-empty heaps if and only if $e_j = 1$. So, for example DAWSON'S CHESS is the OCTAL game $\mathbf{0.137}$, and KAYLES is the game $\mathbf{0.77}$. Observe that, for OCTAL, the value d_0 equals 0 or 4 according to whether it is allowed (value 4) or not (value 0) to split a heap without removing any token. Variants of OCTAL, where the ruleset allows to split a heap without removing any token, have been considered in the literature, starting from GRUNDY'S GAME in 1939 [10]. An extra condition has been imposed to GRUNDY'S GAME, to avoid trivialities, namely the outcome of a split must result in heaps of non-equal sizes. Hence, GRUNDY'S GAME is not in OCTAL.

The notation for OCTAL has been extended to define an arbitrary instance of (pure) TAKE-AND-BREAK, by giving any integer value to each d_i. According to the binary decomposition of $d_i = 2^0 e_0 + 2^1 e_1 + 2^2 e_2 + 2^3 e_3 \ldots$, each value $e_j \in \{0,1\}$ indicates whether it is allowed to remove i tokens and split the remainder into j non-empty heaps. As an example, in the family of HEXADECIMAL games, each d_i is in $\{0, \ldots, 2^4 - 1\}$ (generally encoded by an hexadecimal notation), indicating that a heap can be split into at most three heaps.

We relate the current work to TAKE-AND-BREAK by considering particular rulesets, without 'take'/removal. To our knowledge, the full class of 'break-without-take' has not yet been studied, at least not anywhere

near the extent of the folklore results and conjectures on pure subtraction games, and other OCTAL games, which were researched extensively through the last couple of decades, e.g. [7, 3].

4.2 Regularities in TAKE-AND-BREAK games

Given a TAKE-AND-BREAK game, its $\mathcal{G}$-sequence is the sequence $\mathcal{G}(1), \mathcal{G}(2), \mathcal{G}(3), \ldots$. Finding regularities in $\mathcal{G}$-sequences is a natural objective as it may lead to polynomial-time algorithms that compute the $\mathcal{G}$-values of the game. In particular, periodic behaviors are often observed. A game is said to be *ultimately periodic* with period p and preperiod n_0 if there exist n_0 and p such that $\mathcal{G}(n+p) = \mathcal{G}(n)$ for all $n \geq n_0$. *Pure periodic* games are those for which there is no preperiod; *i.e.* $n_0 = 0$.

For example, it is well known (see [15], Chapter 4, Theorem 2.4 and 2.5) that all finite SUBTRACTION games are ultimately periodic, and a periodicity test has been shown in [1] (see Chapter 7, Corollary 7.34). For OCTAL, the behavior of the $\mathcal{G}$-sequences is far from fully understood, even for very short (finite) rule descriptions. Guy famously asked whether every finite OCTAL game is ultimately periodic. Nowadays, one might lean towards that this is not the case, but it seems to be a very hard problem to understand fully. For example, there are 79 non-trivial OCTAL games with at most 3 octal-digits among which only 14 have been solved [7]. So, in this sense, OCTAL seems perhaps more complicated than CUT[9].

Out of the games that were proved to satisfy this conjecture, one can mention **0.106**, **0.165** and **0.454**. In some cases, the values of the period and the preperiod are huge (e.g. **0.454** has a period of 60620715 and a preperiod of 160949019). On his webpage [7], Flammenkamp maintains a list of octal games with known and unknown periodicities; the game 0.106 has period length 328226140474 and a preperiod of length 465384263797.

J. P. Grossman [8] recently contributed to some "parallelization strategies" and other optimizations that accelerate the computation of Sprague-Grundy values of the game **0.6**, also called OFFICERS, by nearly 60 times of the previous records. His implementation computed over 5 million values per second, a total of more than 140 trillion values over a course of 18 months, without finding any period. And note that OFFICERS is a single digit octal game (!). As was shown in Winning Ways [3], to prove periodicity it suffices to establish that an octal game contains only finitely many $\mathcal{G}$-values. But the problem is that extensive computations reveal the "sparse value phenomenon", and to our knowledge there is no theory yet to handle them. Grossman's approach exploits the fact that sparse values are extremely non-dense, and correctly used, this can dramatically speed up computation, searching for possible regularity (and his method can be used for other games as well). OFFICERS is the only single digit octal game for which Richard K. Guy's original question/conjecture remains unanswered. Somewhat surprisingly, the invariant ditto, the game **0.7** where, in addition, a heap with a single bean may be removed, reverts to "she-loves-me-she-loves-me-not".

As explained in [13], some hexadecimal games also satisfy these properties of normal periodicity (e.g. **0.B3**, **0.33F**). In addition, other types of behavior have been exhibited for hexadecimal games, namely *arithmetic-periodicity*. A take-and-break game is *ultimately arithmetic-periodic* with period p, saltus s, and preperiod n_0 if there exist three integers n_0, p and s such that its $\mathcal{G}$-sequence satisfies $\mathcal{G}(n+p) = \mathcal{G}(n) + s$ for all $n \geq n_0$; if $n_0 = 0$, the sequence is *purely arithmetic-periodic*.

If (strict) arithmetic-periodicity never occurs in OCTAL [2], it makes sense in the context of HEXADECIMAL, where the Sprague-Grundy values may not be bounded. For example, the games **0.13FF** or **0.9B** are proved to be ultimately arithmetic-periodic with period 7 and saltus 4 in [13]. Note that normal and arithmetic-periodicities are not the only kinds of regularities that have been detected in hexadecimal games. In [1], the game **0.205200C** is said to be *sapp-regular*, which means that the $\mathcal{G}$-sequence is an interlacing of two periodic subsequences with an arithmetic-periodic one; this behavior occurs also in variants of octal games with pass moves [12]. In [13], the game **0.123456789** satisfies $\mathcal{G}(2m-1) = \mathcal{G}(2m) = m-1$, except $\mathcal{G}(2^k + 6) = 2^k - 1$. In [9], Grossman and Nowakowski use the notion of *ruler regularity* that arises in the hexadecimal games $\mathbf{0.20} \cdots \mathbf{48}$ with an odd number of intermediate 0s; roughly speaking, it corresponds to a kind of arithmetic-periodic sequence where new terms are regularly introduced that double the length of the apparent period.

For a better understanding of TAKE-AND-BREAK, the question of how to detect a possible regularity using just a small number of computations is paramount. For example, the folklore SUBTRACTION

---

[9] Recall that CUT is necessarily invariant, but octal games are typically non-invariant, by setting for example the 2^0 component to 0 for some digit.

periodicity theorem (see for example Chapter 4 of [15]) relies on that, for a given finite SUBTRACTION game S, it suffices to find a repetition of $\max(S)$ consecutive Sprague-Grundy values, to establish ultimate periodicity. Concerning OCTAL, there is a similar result that has been extensively used to prove the ultimate periodicity of some $\mathcal{G}$-sequences [15]:

Theorem 16 (OCTAL Periodicity Test)**.** *Let G be an octal game $\mathbf{d_0}.\mathbf{d_1} \cdots \mathbf{d_k}$ of finite length k. If there exist $n_0 \geq 1$ and $p \geq 1$ such that*

$$\mathcal{G}(n+p) = \mathcal{G}(n) \ \forall n \in \{n_0, \ldots, 2n_0 + p + k - 1\},$$

then G is ultimately periodic with period p and preperiod n_0.

Such kind of testing properties have also been considered for HEXADECIMAL. In [2], Austin yields a first set of conditions to guarantee the ultimate arithmetic-periodicity of HEXADECIMAL with a saltus equal to a power of 2. An extension was later given by Howse and Nowakowski [13] for HEXADECIMAL having an arbitrary saltus. In both cases, several types of computations must be done. In particular, the arithmetic-periodicity must be checked on a range of values much larger than in Theorem 16 (at least seven times the expected period).

4.3   The partition game CUT as a pure-break game

The partition game CUT is a pure-break game in two senses. The first meaning is that they have no special rule which restricts the type of breaking possible, in contrast with for example GRUNDY'S GAME (a split must result in unequal heap sizes), COUPLES ARE FOREVER (any split into two parts is possible, except if the heap is H_2). Secondly there is no "taking", *i.e.* no subtraction of tokens is possible at any stage of play.

As noted already in going from OCTAL to HEXADECIMAL, allowing a heap to be split into three parts may significantly change the behavior of the $\mathcal{G}$-sequence. In the current paper, we have explored how the $\mathcal{G}$-sequences behave when increasing the number of possible splits of a heap. As one might expect that the complexity of this generalization also increases accordingly, we have chosen to focus on the arguably most natural class of break games, namely *break-without-take*, *i.e.* games where it is never allowed to remove tokens from the disjunctive sum of heaps. The total number of beans in the heaps remain the same until the end of play. If the game starts with a large number n, then the game proceeds by finding finer and finer partitions of this number, until no more refinement is possible.

Theorem 2 means that partition games are somehow closer to HEXADECIMAL than OCTAL. This is not so surprising, by the possibility of an extra cut in HEXADECIMAL. Moreover, we note that partition games form particular instances of TAKE-AND-BREAK.

Proposition 17. *The partition game $G(\mathcal{C})$ played on a heap of size n is equivalent to TAKE-AND-BREAK $\mathbf{0}.\mathbf{d_1}\mathbf{d_2}\cdots$ played on a heap of size $n - 1$, where $d_c = 2^{c+2} - 1$ if $c \in \mathcal{C}$, and $d_c = 0$ otherwise. That is, decompose d_0 in binary, so that $d_0 = \sum e_i 2^{i+1}$. Then, for all $n > 0$, $\mathbf{d_0}.\mathbf{00}\cdots(H_n) = \mathbf{0}.\mathbf{d_1}\mathbf{d_2}\cdots(H_{n-1})$, where $d_i = 2^{i+2} - 1$ if $e_i = 1$ and where $d_i = 0$ if $e_i = 0$.*

Proof. Interpret the game $G(\mathcal{C})$ played on a heap of size n as a game on a path P_n of size n as follows. A k-cut consists in removing k edges from the path, leaving $k + 1$ non-empty smaller paths. Now consider the line graph of P_n (*i.e.* the graph where edges are replaced by vertices, and two vertices are adjacent if the corresponding edges where incident in the initial graph). Clearly, this line graph is a path of size $n - 1$. A k-cut in P_n thus consists in removing k vertices from the line graph. Hence $G(C)$ is equivalent to a take-and-break game $\mathbf{0}.\mathbf{d_1}\mathbf{d_2}\ldots$ on the path of size $n - 1$. As there is no constraint on the edges removed from P_n, there is no constraint on the corresponding vertices in the line graph. Hence, if $k \in \mathcal{C}$, the code of each d_k is the maximal possible value, $2^0 + 2^1 + \cdots + 2^{k+1} = 2^{k+2} - 1$, *i.e.* remove k vertices and leave any number of heaps $\leq k + 1$. $\qquad\square$

Proposition 17 shows that $G(\{1, 2\})$ is equivalent to HEXADECIMAL $\mathbf{0.7F}$. Applied to the game COUPLES-ARE-FOREVER, the same analysis proves that it is equivalent to OCTAL $\mathbf{0.6}$, *i.e.* OFFICERS; moreover a similar argument shows that DAWSON'S CHESS has two equivalent representations $\mathbf{0.137}$ and $\mathbf{0.4}$ (with two additional tokens), and so on.

This discussion concerned several examples of extremely short TAKE-AND-BREAK rulesets with great complexity. By relating to the case of CUT one might guess that in general, the complexity of the $\mathcal{G}$-sequence does not increase with $\max(\mathcal{C})$. Indeed in all case of analysis in this paper, except one, namely $\mathcal{C} = \{1,2\}$, the $\mathcal{G}$-sequence is either periodic or arithmetic-periodic. Let us conclude the paper, with related open questions.

5 Conclusion and perspectives

The results from Sections 2 and 3 are summarized in Table 2. The games are partitioned into three families: those for which periodicity or arithmetic-periodicity is proved, and those for which we used two or three conditions of the AP-test, respectively.

	The ruleset $\mathcal{C}$	Sequence	Theorem
Solved	$\{1, c_2, \ldots\}$ (c_i odd)	$(0,1)$ $(+0)$	Proposition 3
	$\{c_1, \ldots\}$ ($c_1 > 1$)	$(0)^{c_1}$ $(+1)$	Proposition 6
	$\{1, 2, 3, c_4, \ldots\}$	(0) $(+1)$	Proposition 7
	$\{1, 3, 2k\}$ ($k \geq 1$)	$(0,1)^c$ $(+2)$	Proposition 8
AP1 and AP2	$\{1, 2c, 2c' + 1, c_1, \ldots, c_k\}$		Proposition 13
	$\{1, 2c\}$ ($c \geq 2$)		Corollary 15
AP1, AP2 and AP3	$\{1, c_1, \ldots, c_k\}$ (c_i even, $k \geq 1$)		Theorem 9

Table 2: Some partition games.

Among the families that are not solved, all of our computations on particular examples have shown ultimate arithmetic-periodic behaviors, except one, namely $G(\{1,2\})$. This game has a Sprague-Grundy sequence with a lot of visible regularity but some surprising 'drop-out' values, as shown in Figure 1 (in the second picture the drop-outs have been highlighted).
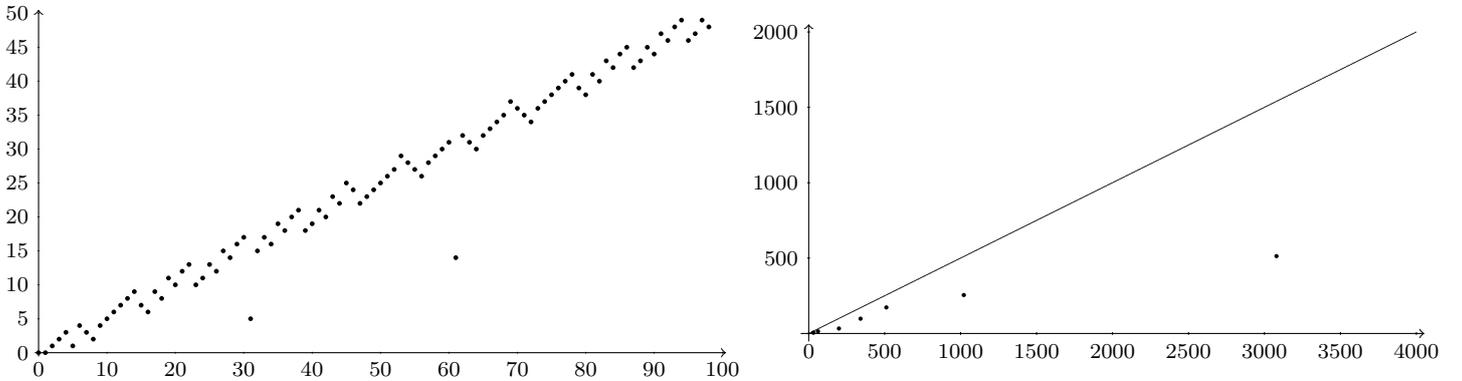

Figure 1: The Sprague-Grundy sequence of $G(\{1,2\})$ for $n \leq 100$ and $n \leq 4000$.

In view of our computations, we propose the following problem.

Open Problem 2. Is it true that every game $G(\mathcal{C})$ with $\mathcal{C} \neq \{1,2\}$ has a Sprague-Grundy sequence that is either ultimately periodic or ultimately arithmetic-periodic?

A first step to understand this problem would be to justify (or disprove) the following conjecture.

Conjecture 1. *Every instance $\mathcal{C}$ of* CUT *for which $\{1,2\} \not\subset \mathcal{C}$ is (ultimately) arithmetic-periodic.*

For some games, the conjecture holds but where a non-computer justification becomes at the best tedious (e.g. $\mathcal{C} = \{1, 2, 7\}$). A general proof is probably hard to obtain, and meanwhile this motivates the testing conditions. The AP-test is a rather short computation to justify arithmetic-periodicity of a game; for even more simplicity, we wonder whether the condition AP3 could be removed from the test.

Open Problem 3. Do AP1 and AP2 imply AP3?

Note that removing AP3 from the AP-test would not have much impact on the computational complexity of the test, but this would make the AP-test closer to its counterparts for other take-and-break games.

Concerning the ultimate property of the sequence, we have seen that in almost all considered cases, there is no preperiod. However, the case $\mathcal{C} = \{1, 2, 8\}$ suggests that some games have one. Therefore, the question of finding an equivalent of the AP-test for these games makes sense.

Open Problem 4. Find a generalization of the AP-test to prove ultimate arithmetic-periodicity of partition games.

As indicated by Figure 1, the game $G(\{1, 2\})$ leaves a couple of questions open. What is the number of occurrences of each Sprague-Grundy value of $G(\{1, 2\})$? We already know from Theorem 2 that every Sprague-Grundy value appears at most twice in the sequence of $G(\{1, 2\})$ (apply Theorem 2 with $c = 2$).

Open Problem 5. Does each Sprague-Grundy value appear at least once in the sequence of $G(\{1, 2\})$? More precisely, does each $\mathcal{G}$-value appear exactly twice in the sequence of $G(\{1, 2\})$?

Open Problem 6. What is the behavior of the Sprague-Grundy sequence of $G(\{1, 2\})$?

Recall that from Proposition 17, this game is equivalent to HEXADECIMAL **0.7F** (that is also claimed unsolved), with rules: remove one token and leave no heap, remove one token and leave one heap, remove one token and leave two heaps, remove two tokens and leave no heap, remove two tokens and leave one heap, remove two tokens and leave two heaps, or remove two tokens and leave three heaps. The CUT description is here more concise, (without removing any token) split your heap in two or three parts, and with equivalent behavior.

On the other hand a full generalization of hexadecimal games includes as a subclass CUT, but we feel perhaps the general class is too large, and further classification of partition games appears a more approachable (sub-)goal. Another interesting direction would be to pursue Grundy's idea to disallow symmetric partitions (but with no removal), but we have not yet looked into that.

Acknowledgement

References

[1] M. Albert, R. J. Nowakowski & D. Wolfe, *Lessons in Play: An Introduction to Combinatorial Game Theory*, A.K. Peters (2007).

[2] R. B. Austin, *Impartial and Partizan Games*, M.Sc. Thesis, The University of Calgary (1976).

[3] E. R. Berlekamp, J. H. Conway, & R. K. Guy. *Winning ways for your mathematical plays (Vol. 1).* Natick, MA: AK Peters (2001).

[4] I. Caines, C. Gates, R. K. Guy & R. J. Nowakowski, Periods in taking and partitioning games, *Amer. Math. Monthly* **106** (1999) 359–361.

[5] T. R. Dawson (1935) "Caissa's Wild Roses", in *Five Classics of Fairy Chess*, Dover Publications Inc., New York (1973).

[6] E. Duchêne & M. Rigo, Invariant games, *Theoretical Computer Science* **411** (2010) 3169–3180.

[7] A. Flammenkamp, Sprague-Grundy values of some octal games, `wwwhomes.uni-bielefeld.de/achim/octal.html`.

[8] J. P. Grossman, Searching for Periodicity in Officers, *Games of No Chance 5*, MSRI Publications, Cambridge University Press, Cambridge, volume 70 (2019).

[9] J. P. Grossman & R. J. Nowakowski, A ruler regularity in hexadecimal games, *Games of No Chance 4*, MSRI Publications, Cambridge University Press, Cambridge (2013).

[10] P. M. Grundy. Mathematics and games. *Eureka* **2** (1939) 6–9.

[11] R. K. Guy and C. A. B. Smith "The G-values of various games" *Proc. Camb. Phil. Soc.* 52, (1956) 512–526.

[12] D. G. Horrocks & R. J. Nowakowski, Regularity in the $\mathbb{G}$-sequences of octal games with a pass, *INTEGERS* **3** (2003), G1.

[13] S. Howse & R. J. Nowakowski, Periodicity and arithmetic-periodicity in hexadecimal games, *Theoretical Computer Science* **313** (2004) 463–472.

[14] R. J. Nowakowski, Unsolved problems in combinatorial games, *Games of No Chance 5*, MSRI Publications, Cambridge University Press, Cambridge, volume 70 (2019).

[15] A. N. Siegel, *Combinatorial Game Theory*, San Francisco, CA (2013).

[16] R. P. Sprague, Über mathematische Kampfspiele, *Tôhoku Math. J.* 41, 438–444, (1935-36), ZBl, 13, 290