Nimhoff Games

AVIEZRI S. FRAENKEL AND MORDECHAI LORBERBOM

Department of Applied Mathematics & Computer Science,
The Weizmann Institute of Science, Rehovot 76100, Israel

Communicated by the Managing Editors
Received December 15, 1988.

Nimhoff designates a class of two-player perfect-information combinatorial games. Let \( R \subseteq (\mathbb{Z}^0)^n \) be an \( n \)-ary relation. Given \( n \) piles of tokens, two players move alternately in Nimhoff (=Nimhoff\( (R) \)). The moves are of two types: a player may remove any (positive) number of tokens from a single pile, or he may take \( a_i \) tokens from the \( i \)-th pile for \( i = 1, \ldots, n \), such that \( R(a_1, \ldots, a_n) \). The player first unable to move is the loser; his opponent the winner. Known examples are Nim (when \( R \) is the empty relation), which is an easy game; and Wythoff's game \((n = 2 \text{ and } R(a_1, a_2) \text{ if } a_1 = a_2)\), whose Sprague-Grundy function computation is not known to be polynomial. These games give Nimhoff its name. Our main goal is to define and analyze games lying in between, aimed at bridging the complexity gap between Nim and Wythoff's game.

\[ \text{097-3165/91 $3.00$} \]

\[ \text{Copyright \( \odot \) 1991 by Academic Press, Inc.} \]

All rights of reproduction in any form reserved.
Notations. 1. Throughout this paper the binary encoding of a nonnegative integer \( h \) is denoted by \( \ldots, h^2, h^1, h^0 \), where \( h = \sum_{i \geq 0} h^i 2^i \) (\( h^i \in \{0, 1\}, i \geq 0 \)).

2. For \( a \in \mathbb{Z}^+ \) and \( h \in \mathbb{Z}^+ \), denote by \( a(h) \) the unique integer \( r \in [0, h - 1] \), such that \( r \equiv a \pmod{h} \).

3. For \( x, y \in \mathbb{Z}^0 \), denote by \( x \oplus y \) the Nim-sum of \( x \) and \( y \) (\( x \oplus y = \sum_{i \geq 0} [(x^i + y^i)(2)] 2^i \), see, e.g., [1, 2, or 3]), and let
\[
\sum_{i=1}^{n} x_i = x_1 \oplus x_2 \oplus \cdots \oplus x_n.
\]

With these notations we can write a formula to compute the Sprague–Grundy function (\( g \)-function for short) for any position of Nim, namely,
\[
g(b_1, \ldots, b_n) = \sum_{i=1}^{n} b_i,
\]
where the winner needs only to move to \( (b_1, \ldots, b_n) \) with \( g(b_1, \ldots, b_n) = 0 \), which can be computed in polynomial time.

On the other hand, we have Wythoff’s game. Wythoff’s game is played on two piles of tokens, and the player may take any (positive) number of tokens from a single pile, or an equal (positive) number of tokens from both piles. There is a polynomial algorithm for constructing all \( (b_1, b_2) \) satisfying \( g(b_1, b_2) = 0 \) (see Yaglom and Yaglom [6]); but despite much effort, no polynomial algorithm for deciding whether \( g(b_1, b_2) = k \) for any \( k > 0 \), is known. In particular, no polynomial algorithm is known for computing the sum of Wythoff games, and this seems to contribute to the fact that no natural way to generalize Wythoff’s game to more than two piles is known. For other generalizations of Wythoff’s game see Fraenkel and Borosh [4].

Between these extreme cases, there are classes of Nimhoff games whose strategies are not as easy as the strategy for Nim, but are still polynomial. This paper deals with some such classes. In Section 2 we define \( \text{Nimdi} \) to be the class of relations with the same strategy as Nim, i.e., \( g(b_1, \ldots, b_n) = \sum'_{i=1} b_i \). We give a sufficient condition for Nimdi. The conjecture is that this condition is also necessary; we prove it only for special cases.

When \( n = 2 \), it is useful to look at the grid of the \( g \)-values and analyze its structure. A grid of a game is a table of \( g \)-values, namely \( g(b_1, b_2) \) appears in row \( b_1 \) and column \( b_2 \). An example of a Nimhoff game whose grid helps us to find a formula for its \( g \)-function, is the king-rook game (see Gardner [5]). A king-rook is a “fairy-chess” piece, which combines the powers of a king and a rook. The king-rook game starts with a king-rook.
positioned somewhere on a chessboard. Each player at his turn moves the
king-rook either north, west, or northwest. The player who first reaches the
corner wins, his opponent loses. It is easy to see that the king-rook game
is equivalent to Nimhoff(\(R\)) for \(R(1, 1)\). A grid of the king-rook game is
depicted in Fig. 1.

We can see that the grid can be partitioned into \(3 \times 3\) blocks. Every
block is the cyclic permutation of \(3m, 3m + 1, 3m + 2\) \((m \geq 0)\). Moreover, if
we replace every block with the corresponding \(m\), we get the grid of Nim,
which is depicted in Fig. 2 in Section 5. Hence the formula for the \(g\)-func-
tion is \(g(b_1, b_2) = 3(\lfloor b_1/3 \rfloor \oplus \lfloor b_2/3 \rfloor) + (b_1 + b_2)(3)\). A generalization of
this game (cyclic Nimhoff) and a proof can be found in Section 3.
In Section 4 we deal with balanced Nimhoff with powers of 2, in which
a player can move an equal number \(2^k\) from two piles. The formula is
\(\mathcal{N}\)-Nim-sum, which is a modification of the regular Nim-sum, with some
interesting properties.

From Section 5 on, we consider the case \(n = 2\). In Section 5 we introduce
the concept of "maximal binary common suffix" (mats for short), and show
how to compute it efficiently. This concept is useful in Sections 6 and 7 to
make the formulas compact. In Section 6 we present "double cyclic
Nimhoff," which is a variation of cyclic Nimhoff. For example, if we permit
the king-rook to move three positions northwest, it makes the blocks of the
grid to "cycle" again. In Section 7 we deal with "even balanced Nimhoff,"
which is \(R(2l, 2l)\) for a fixed \(l\). A certain relationship between the last two
classes is pointed out at the end of Section 7.

\[
\begin{array}{cccc|cccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\
1 & 2 & 0 & 4 & 5 & 3 & 7 & 8 & 6 & 10 & 11 & 9 \\
2 & 0 & 1 & 5 & 3 & 4 & 8 & 6 & 7 & 11 & 9 & 10 \\
3 & 4 & 5 & 0 & 1 & 2 & 9 & 10 & 11 & 6 & 7 & 8 \\
4 & 5 & 3 & 1 & 2 & 0 & 10 & 11 & 9 & 7 & 8 & 6 \\
5 & 3 & 4 & 2 & 0 & 1 & 11 & 9 & 10 & 8 & 6 & 7 \\
6 & 7 & 8 & 9 & 10 & 11 & 0 & 1 & 2 & 3 & 4 & 5 \\
7 & 8 & 6 & 10 & 11 & 9 & 1 & 2 & 0 & 4 & 5 & 3 \\
8 & 6 & 7 & 11 & 9 & 10 & 2 & 0 & 1 & 5 & 3 & 4 \\
9 & 10 & 11 & 6 & 7 & 8 & 3 & 4 & 5 & 0 & 1 & 2 \\
10 & 11 & 9 & 7 & 8 & 6 & 4 & 5 & 3 & 1 & 2 & 0 \\
11 & 9 & 10 & 8 & 6 & 7 & 5 & 3 & 4 & 2 & 0 & 1 \\
\end{array}
\]

FIG. 1. The grid of the king-rook game.
In most of the classes there are redundant comoves. These are comoves which do not help the loser, and the winner does not have to use them. Formally, if the \( g \)-function of \( \text{Nimhoff}(R \cup (a_1, \ldots, a_n)) \) is identical to the \( g \)-function of \( \text{Nimhoff}(R) \), then we say: (i) the comove \((a_1, \ldots, a_n)\) of \( \text{Nimhoff}(R \cup (a_1, \ldots, a_n)) \) is redundant (with respect to \( \text{Nimhoff}(R) \)); (ii) the redundant comove \((a_1, \ldots, a_n)\) may be adjoined to \( \text{Nimhoff}(R) \) to get the equivalent game \( \text{Nimhoff}(R \cup (a_1, \ldots, a_n)) \). For example, all the comoves in Nimdi are redundant. A relation without redundant comoves is called a nucleus. Throughout the paper we try to identify the redundant comoves and the nuclei of the various classes.

The motivation for this work, triggered by \([S]\), thus stems from the following facts: (i) Nim is very easy. (ii) Wythoff games seem to be very difficult. (iii) The nature of the difficulty is not simply the nondisjunctive move of taking from both piles (see Nimdi below); it is rather the fact that the move \((a, a)\) preserves Nim-sum \(0\). (iv) This observation suggests a large class of games lying in between Nim and Wythoff's game, thus spanning a bridge between them.

2. Nimdi

Nimdi designates the class of Nimhoff games in which the strategy is the same as Nim; i.e., it gives the players more comoves than Nim, but these comoves do not help the loser, and the winner does not have to use these comoves. (The name Nimdi is now clear: Nim in disguise.) We look for conditions on the comove \((a_1, \ldots, a_n)\) such that the game will be in Nimdi.

**Proposition 1.** Let \( k \) be the maximal number such that \( 2^k \mid a_i \) for every \( i \in [1, n] \). If the number of \( a_i \) with \( 2^{k+1} \mid a_i \) is odd, then the comove \((a_1, \ldots, a_n)\) is redundant with respect to Nim.

*Proof.* It suffices to show that removing \((a_1, \ldots, a_n)\) changes the Nim-sum. Indeed, \( 2^k \mid a_i \) implies that there is no carry in the subtraction of \( a_i \) to the right of digital position \( k \). Since the number of 1's in \( \{a_1^k, \ldots, a_n^k\} \) is odd, the \( k \)th bit is complemented in the Nim-sum of the differences.

A special case of Proposition 1 is when \( k = 0 \). Then Proposition 1 deals with an odd number of odd numbers. For example, the rook-knight game (see \([S]\)) is the special case \( n = 2 \) and \( \{a_1, a_2\} = \{1, 2\} \).

Proposition 1 gives a sufficient condition on \((a_1, \ldots, a_n)\) to be redundant. The interesting question is, of course, whether this condition is also necessary in the following sense: Given \((a_1, \ldots, a_n)\) which do not satisfy the hypothesis of Proposition 1, i.e., \( 2^k \mid a_i \) for every \( i \in [1, n] \) and the number of \( a_i \) with \( 2^{k+1} \mid a_i \) is positive and even. Are there \((b_1, \ldots, b_n)\) with \( b_i \geq a_i \),...
i = 1, ..., n, such that $\sum_{i=1}^{n} b_i - \sum_{i=1}^{n} (b_i - a_i)$? The conjecture is yes; we proved it only for a special "regular" case.

**Notation.** Let $x$ be a positive integer. Denote by $\eta(x)$ the position of the rightmost 1 in $x^0, x^1, ...$; that is to say $2^{\eta(x)} \mid x$ but $2^{\eta(x)+1} \nmid x$.

**Proposition 2.** Let $a_1, ..., a_n$ be nonnegative integers satisfying the following condition: For every $l \geq 0$, the number of positive $a_i$ with $\eta(a_i) = l$ is even. Then there are $b_1, ..., b_n$ with $b_i \geq a_i$ for $i \in [1, n]$ such that $\sum_{i=1}^{n} b_i = \sum_{i=1}^{n} (b_i - a_i)$.

**Proof:** We show it first for the case $n = 2$ and $\eta(a_1) = \eta(a_2) = 0$ (i.e., two odd numbers). Without loss of generality assume $a_1 > a_2$. Write $a_1 = 2d_1 + 1$, $a_2 = 2d_2 + 1$. Let $u = \lfloor \log_2 d_1 \rfloor + 1$. Then $b_1 = 2^u + d_1$ and $b_2 = 2^u + d_2$ are the desired numbers. Indeed, since $2^u > d_1 \geq d_2$,

$$(b_1 - a_1) \oplus (b_2 - a_2) = (2^u - 1 - d_1) \oplus (2^u - 1 - d_2) = d_1 \oplus d_2 = b_1 \oplus b_2,$$

where the second equality comes from the fact that $2^u - 1 - d_i$ is the 1's complement of $d_i$, $i = 1, 2$.

In the general case divide the positive $a_i$s into pairs such that two numbers $a_i$ and $a_j$ are in a pair if $\eta(a_i) = \eta(a_j) = r$. Consider the odd numbers $a_i/2^r$ and $a_j/2^r$. There are $b_s$ and $b_t$ such that $b_s \oplus b_t = (b_s - a_i/2^r) \oplus (b_t - a_j/2^r)$. Take $b_s = 2^rb_s$ and $b_t = 2^rb_t$; do it for every pair and we are done. \hfill \blacksquare

3. **Cyclic Nimhoff**

Cyclic Nimhoff is the class of Nimhoff games in which the relation $R$ is of the form $R(a_1, ..., a_n)$ if $0 < \sum_{i=1}^{n} a_i < h$, where $h$ is a fixed positive integer.

Special cases: (i) Nim is the case $h = 1$ or 2; (ii) the king-rook game (see Section 1) is the case $n = 2$ and $h = 3$; (iii) similarly, a king-rook-knight game can be defined, which corresponds to the case $n = 2$ and $h = 4$.

**Lemma 1.** Let $b_1, ..., b_n$ be nonnegative integers, and let $h \in \mathbb{Z}^+$. For any $r$ satisfying $0 < r \leq (\sum_{i=1}^{n} b_i)(h)$, there are $a_1, ..., a_n$ with $0 \leq a_i \leq b_i(h)$ ($1 \leq i \leq n$), such that $\sum_{i=1}^{n} a_i = r$.

**Proof:** Since

$$0 < r \leq \left( \sum_{i=1}^{n} b_i \right)(h) \leq \sum_{i=1}^{n} b_i(h),$$
it follows that there is \( j \in [1, n] \) such that
\[
\sum_{i=1}^{j-1} b_i(h) < r \leq \sum_{i=1}^{j} b_i(h).
\]
Take \( a_i = b_i(h) \) for \( i < j \), \( a_j = r - \sum_{i=1}^{j-1} b_i(h) \), and \( a_i = 0 \) for \( i > j \). Since
\[
r \leq \sum_{i=1}^{j} b_i(h) = \sum_{i=1}^{j-1} b_i(h) + b_j(h),
\]
we have \( 0 < a_j \leq b_j(h) \); and
\[
\sum_{i=1}^{n} a_i = \sum_{i=1}^{j-1} a_i + a_j + \sum_{i=j+1}^{n} a_i
\]
\[
= \sum_{i=1}^{j-1} b_i(h) + r - \sum_{i=1}^{j-1} b_i(h) + 0
\]
\[
= r.
\]

An \textit{option} of a position \( u \) of a game is any position reachable from \( u \) in one move. For every \( w > 0 \) let \( \bar{w} = \lfloor w/h \rfloor \).

The following result gives a formula to compute the \( g \)-function for any position of cyclic Nimhoff.

**Theorem 1.** The \( g \)-function of a position \((b_1, \ldots, b_n)\) of cyclic Nimhoff is
\[
g(b_1, \ldots, b_n) = h\left(\sum_{i=1}^{n} b_i\right) + \left(\sum_{i=1}^{n} b_i\right)(h).
\]

Note the combination of Nim-sum and ordinary sum, which is somewhat reminiscent of Welter's game [3, Chap. 13].

**Proof.** Denote by \( G(b_1, \ldots, b_n) \) the right-hand side of (1). In order to prove that \( g(b_1, \ldots, b_n) = G(b_1, \ldots, b_n) \), we have to show the following two properties:

(A) \( G(b_1, \ldots, b_n) \notin \{G(b'_1, \ldots, b'_n) : (b'_1, \ldots, b'_n) \text{ is an option of } (b_1, \ldots, b_n)\} \).

(B) For any \( G' < G(b_1, \ldots, b_n) \), there is an option \((b'_1, \ldots, b'_n)\) of \((b_1, \ldots, b_n)\), such that \( G' = G(b'_1, \ldots, b'_n) \).

These two claims jointly imply \( g(b_1, \ldots, b_n) = G(b_1, \ldots, b_n) \).

(A) Let \((b'_1, \ldots, b'_n)\) be an option of \((b_1, \ldots, b_n)\), and assume that \( G(b_1, \ldots, b_n) = G(b'_1, \ldots, b'_n) \). From the definition of \( G \) we have
Since \((b'_1, ..., b'_n)\) is an option of \((b_1, ..., b_n)\), there are two possibilities:

(I) There is \(k \in [1, n]\) such that \(b'_k < b_k\) and \(b'_i = b_i\) for \(i \neq k\). Then (3) implies \(b'_k \equiv b_k \pmod{h}\). But then we get \(b'_k < b_k\), a contradiction to (2), since \(a' < a\) implies \(a \oplus b \neq a' \oplus b\).

(II) There are \(a_1, ..., a_n\) with \(0 < \sum_{i=1}^n a_i < h\), such that \(b'_i = b_i - a_i\) for \(i \in [1, n]\). Since \(\sum_{i=1}^n a_i \neq 0 \pmod{h}\), we get an immediate contradiction to (3).

(B) Let \(G = G(b_1, ..., b_n)\). For given \(G' < G\), there are two cases:

(I) \(\overline{G'} = \overline{G}\). We have \(0 < G - G' \leq G(h) = (\sum_{i=1}^n b_i)(h)\). By Lemma 1, there are \(a_1, ..., a_n\) with \(0 \leq a_i \leq b_i(h)\), such that \(\sum_{i=1}^n a_i = G - G'\). Take \(b'_i = b_i - a_i\) for \(i \in [1, n]\). We will show that

\[
G' = h \left( \sum_{i=1}^n b'_i \right) + \left( \sum_{i=1}^n b'_i \right) (h). \tag{4}
\]

Indeed,

\[
\sum_{i=1}^n b'_i = \sum_{i=1}^n \left[ \frac{b'_i}{h} \right] = \sum_{i=1}^n \left[ \frac{b_i - a_i}{h} \right] = \sum_{i=1}^n \left[ \frac{b_i}{h} \right] = \sum_{i=1}^n b'_i = G = G';
\]

\[
\left( \sum_{i=1}^n b'_i \right) (h) = \left( \sum_{i=1}^n (b_i - a_i) \right) (h)
= \left( \sum_{i=1}^n b_i - (G - G') \right) (h)
= \left( \sum_{i=1}^n b_i \right) (h) - (G(h) - G'(h))(h)
= G'(h).
\]

(II) \(\overline{G'} < \overline{G}\). Since \(\sum_{i=1}^n b_i = \overline{G}\), a basic property of Nim-sum implies that there is \(k \in [1, n]\), and \(\overline{A} < b_k\) such that \(\overline{A} \oplus \sum_{i \neq k} b_i = \overline{G'}\). In this case take \(b'_k = h\overline{A} + (G' - \sum_{i \neq k} b_i)(h)\), and \(b'_i = b_i\) for all \(i \neq k\). Note that \(b'_k < b_k\), so \((b'_1, ..., b'_n)\) is an option of \((b_1, ..., b_n)\). Again we have to show that (4) holds. Indeed,
\[
\sum_{i=1}^{n} b'_i = b'_k \oplus \sum_{i \neq k} b_i = \bar{A} \oplus \sum_{i \neq k} b_i = G';
\]

\[
(\sum_{i=1}^{n} b'_i)(h) = \left( b'_k(h) + \left( \sum_{i \neq k} b'_i \right)(h) \right)(h) = \left( \left( G' - \sum_{i \neq k} b_i \right)(h) + \left( \sum_{i \neq k} b_i \right)(h) \right)(h) = G'(h).
\]

Redundant comoves and nucleus. \( R \) is a nucleus because for every \((a_1, ..., a_n) \in R\) take \((b_1, ..., b_n) = (a_1, ..., a_n)\), and the winner must use this move, otherwise he loses.

Redundant comoves may be adjoined; namely, any \((a_1, ..., a_n)\) with \(\sum_{i=1}^{n} a_i \not\equiv 0 \pmod{h}\). The formula for \( g \) will still be (1). For the proof of part (B) we do not need this additional option. The proof of part (A) remains invariant under this change. Thus, for example, a game equivalent to the king-rook game is: remove from a single pile, or a total of \( k \) tokens from both piles, provided \( k \not\equiv 0 \pmod{3} \).

4. Balanced Nimhoff with Powers of 2

\( 2^k \)-balanced Nimhoff designates the class of Nimhoff games in which a player may take \( 2^k \) tokens from two piles. Formally, let \( k \) be a fixed positive integer, and define \( R(a_1, ..., a_n) \) if \( a_s = a_t = 2^k \) for some \( s \neq t \) and \( a_i = 0 \) for every \( i \neq s, t \).

Let \( w \) be any nonnegative integer. Denote by \( i(w) \) the least significant bit complementation function: \( i(w) = w \oplus 1 \). If \( a \) and \( b \) are nonnegative integers then their \( k \)-Nim-sum is defined by \( a \otimes b = a \oplus b \oplus a^kb^k \). In other words, the \( k \)-Nim-sum of \( a \) and \( b \) is \( a \oplus b \) unless \( a^k = b^k = 1 \), in which case it is \( i(a \oplus b) \). (Note that the regular Nim-sum is not a special case of \( k \)-Nim-sum.) It is easy to see that \( \mathcal{Z}^0 \) is a commutative group under \( k \)-Nim-sum with unit 0, and the inverse of \( a \) is \( a \) (if \( a^k = 0 \)); \( i(a) \) (if \( a^k = 1 \)). Therefore we can define

\[
\sum_{i=1}^{n} x_i = x_1 \oplus x_2 \oplus \cdots \oplus x_n.
\]

**Lemma 2.** Given nonnegative integers \( x_1, ..., x_n \). Let \( w \) be the number of 1's in \( (x_1^k, x_2^k, ..., x_n^k) \). If \( w \equiv 0 \) or 1 (mod 4) then \( \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} x_i \), else \( \sum_{i=1}^{n} x_i = i(\sum_{i=1}^{n} x_i) \).
Proof. The definition of $k$-Nim-sum implies that all the bits of $\sum_{i=1}^{n} x_i$ and $\sum_{i=1}^{n} x'_i$ are identical except, possibly, the lowest one, which may be complemented. Every two 1’s in the $k$th column complement the lowest bit. So the complementation of the lowest bit depends upon the parity of the number of pairs of 1’s in column $k$: If there is an even numbers of pairs ($w \equiv 0$ or $1 \pmod{4}$) then there is no complementation; otherwise—there is.

The following result shows that the $k$-Nim-sum is precisely the $g$-function of $k$-Nimhoff.

Theorem 2. The $g$-function of a position $(b_1, \ldots, b_n)$ of $2^k$-balanced Nimhoff is

$$g(b_1, \ldots, b_n) = \sum_{i=1}^{n} b_i.$$ (5)

Proof. Denote by $G(b_1, \ldots, b_n)$ the right-hand side of (5).

(A) Let $(b'_1, \ldots, b'_n)$ be an option of $(b_1, \ldots, b_n)$. There are two possibilities:

(I) There is $t \in [1, n]$ such that $b'_t < b_t$ and $b'_t = b_t$ for $i \neq t$. Since $2^0$ is a group under $k$-Nim-sum, it follows that

$$\sum_{i=1}^{n} b_i = \sum_{i=1}^{n} b'_i.$$

(II) There are $s, t \in [1, n]$ with $s \neq t$, such that $b'_s = b_s - a_s$, $b'_t = b_t - a_t$, and $b'_i = b_i$ for $i \neq s, t$, where $a_s = a_t = 2^k$. Let $c = b_s \oplus b_t$ and $c' = b'_s \oplus b'_t$. It suffices to show that $c \neq c'$. There are two cases:

1. $b'_s = b'_t$. Then $b'_s = b'_t \neq b^k_s$. Since $a_s = a_t = 0$, it follows that $c^0 \neq c'^0$.

2. $b'_s \neq b'_t$. Without loss of generality $b'_s = 0$ and $b'_t = 1$. There is $r > k$ such that $b'_s = 1$ and $b'_t = 0$. But $b'_s = b'_t$. Hence $c' \neq c''$.

(B) Let $G = G(b_1, \ldots, b_n)$, and assume that $G' < G$. Since Lemma 2 implies $G' < \sum_{i=1}^{n} b_i$, there are two cases:

(I) $G' < \sum_{i=1}^{n} b_i$. By the basic property of Nim-sum, there is $t \in [1, n]$, and $b_t < b'_t$, such that $b_t \oplus \sum_{i \neq t} b_i = G'$. If $b, \sum_{i \neq t} b_i = G'$, take $b'_i = b_i$, and $b'_t = b_t$ for every $i \neq t$ and we are done. Otherwise, by Lemma 2, the two numbers $b_t \oplus \sum_{i \neq t} b_i$ and $G'$ differ in the lowest bit. So take $b_t' = i(b_t)$, and $b'_t = b_t$ for every $i \neq t$. Then $G' = \sum_{i=1}^{n} b'_i$. Moreover, $b'_t < b_t$, because $b'_t = b_t$ implies

$$G' = \sum_{i=1}^{n} b'_i = \sum_{i=1}^{n} b_i = G.$$
(II) $G' = \sum_{i=1}^{n} b_i$, so we have $G' = G - 1$. By Lemma 2, the number of 1's in $\{b_1, b_2, ..., b_n\}$ is $\equiv 2$ or $3 \pmod{4}$. Hence there are $s, t \in [1, n]$ with $s \neq t$, such that $b_s^* = b_t^* = 1$. Take $b'_s = b_s - 2^k$, $b'_t = b_t - 2^k$, and $b'_i = b_i$ for $i \neq s, t$. We have

$$G(b'_1, ..., b'_n) = \sum_{i=1}^{n} (b'_i = \sum_{i=1}^{n} b_i = G'. 1$$

**Note.** The second type of move falls in the "regular" case of the Nimdi part ($\eta(a_s) = \eta(a_t)$), which can lead to the same Nim-sum (Proposition 2). Here since we have $k$-Nim-sum, it does not help the loser as we saw.

**Definition.** If during a subtraction, there is a borrow from the $k$th bit to the $(k - 1)$th bit, we say that the $k$th bit generates a borrow, and the $(k - 1)$th bit absorbs a borrow. For example, in the subtraction $10 - 01$, the left bit of 10 (the 1) generates a borrow, and the right bit (the 0) absorbs a borrow.

**Redundant comoves and nucleus.** The relation $R$ is clearly a nucleus. As redundant comoves we can adjoin any removal of $a_s$ and $a_t$ such that $a_s^{k+1} = a_t^{k+1}$ but $a_s^k \neq a_t^k$. This fact follows from the following result:

**Lemma 3.** Let $a_s \leq b_s$ and $a_t \leq b_t$. If $a_s^{k+1} = a_t^{k+1}$ and $a_s^k \neq a_t^k$ then $b_s \not\equiv b_t \equiv (b_s - a_s) \not\equiv (b_t - a_t)$.

**Proof.** Using the notations of Theorem 2 part (A) case (II), we have to show that $c \neq c'$. If $c^k \neq c'^k$ we are done; otherwise, exactly one of $b^k_s$ and $b^k_t$ generates a borrow to the $(k - 1)$th bit in subtracting $a_i$ ($i \in \{s, t\}$). Assume without loss of generality that $a_s^k = 0$, $a_t^k = 1$. If $b^k_s$ generates a borrow and $b^k_t$ does not generate a borrow, then $b^k_t$ absorbs a borrow from the $(k + 1)$th bit while $b^k_s$ does not absorb a borrow, so $c^{k+1} \neq c'^{k+1}$. If $b^k_s$ generates a borrow and $b^k_t$ does not generate a borrow, then there are two cases: (i) $b^k_s \neq b^k_t$. Again one of $b^k_s$ and $b^k_t$ (the zero) absorbs a borrow, and the second (the one) does not, so $c^{k+1} \neq c'^{k+1}$; (ii) $b^k_s = b^k_t$. In this case $b^k_s b^k_t \neq b^k_s b^k_t$, so we have $c^0 \neq c'^0$.

A special case of Lemma 3 is when $0 < a_s, a_t < 2^{k+1}$, $a_s \neq a_t$ with $a_s + a_t = 2^{k+1}$. If we adjoin these redundant comoves to the nucleus, we get a relation $R'(a_1, ..., a_n)$ if $a_s + a_t = 2^{k+1}$. It reminds us of cyclic Nimhoff whose relation is also on the sum of the components of the comove. However, unlike cyclic Nimhoff, here we do not permit removing from more than two piles (i.e., $R(a_1, ..., a_n)$ if $\sum_{i=1}^{n} a_i = 2^{k+1}$), since it can lead from $k$-Nim-sum zero to $k$-Nim-sum zero. For example, take $k = 1$; the position $(0, 0, 0, 0)$ is an option of $(1, 1, 1, 1)$. The phenomenon that the
relation can be stated in terms of sum of components, recurs throughout the rest of the paper.

5. Maximum Binary Common Suffix

Notations. Let $a$ and $b$ be nonnegative integers. The maximum binary common suffix of $a$ and $b$, denoted $\text{macs}(a, b)$, is the number whose binary encoding is the common right bits of $a$ and $b$. Formally, if there is $j \geq 0$ such that $a^i = b^i$ for $0 < i < j$ and $a^j \neq b^j$, then $\text{macs}(a, b) = \sum_{i=0}^{j-1} a^i 2^i \left(= \sum_{i=0}^{j-1} b^i 2^i\right)$, otherwise $\text{macs}(a, b) = a(= b)$.

We point out the similarity of the $\text{macs}$ to the "mating function" used by Conway in his analysis of Welter's game [3, Chap. 13].

Let $a$ and $b$ be nonnegative integers, and let $k$ be a positive integer. Denote by $\text{macs}_k(a, b)$ the number of times it is possible to subtract $k$ from $a$ and $b$ without changing the Nim-sum. That is to say,

$$\text{macs}_k(a, b) = \max\{ d : a \oplus b = (a - ik) \oplus (b - ik) \text{ for every } d \}.$$

Recall that we defined Nim-sum only on nonnegative integers.

From now on we will use letters to denote either numbers or their binary encodings. The meaning in each case will be clear from the context.

**Lemma 4.** If $a \oplus b = (a + 1) \oplus (b + 1)$, then $\text{macs}(a + 1, b + 1) = \text{macs}(a, b) + 1 > 0$; and conversely if $\text{macs}(a, b) > 0$, then $a, b > 0$ and $a \oplus b = (a - 1) \oplus (b - 1)$.

**Proof.** If $a \oplus b = (a + 1) \oplus (b + 1)$, then $a$ and $b$ end with the same number, say $l$ ($l \geq 0$), of 1's, i.e., $a^i - b^i = 1$ for $0 < i < l$ and $a^l - b^l = 0$. So, $\text{macs}(a, b)$ also ends with 0, followed on the right by $l$ 1's. On the other hand, $\text{macs}(a + 1, b + 1)$ begins exactly as $\text{macs}(a, b)$ but ends with 1, followed on the right by $l$ 0's, so we have $\text{macs}(a + 1, b + 1) = \text{macs}(a, b) + 1$.

Conversely, if $\text{macs}(a, b) > 0$, then clearly $a, b > 0$. Moreover, $a$ and $b$ end with 1, followed on the right by the same number, say $l$ ($l \geq 0$), of 0's. So $a \oplus b$ ends with $l + 1$ 0's. On the other hand, $a - 1$ begins the same as $a$ but ends with 0, followed on the right by $l$ 1's, and the same for $b - 1$. Hence $(a - 1) \oplus (b - 1)$ begins the same as $a \oplus b$ and also ends with $l + 1$ 0's, and thus we have $a \oplus b = (a - 1) \oplus (b - 1)$.

**Corollary 1.** For every $a, b \geq 0$, $\text{macs}(a, b) = \text{macs}_1(a, b)$.

**Proof.** By induction on $m = \text{macs}_1(a, b)$.

(i) $m = 0$. If $\text{macs}(a, b) > 0$ then according to Lemma 4, $a \oplus b = (a - 1) \oplus (b - 1)$, so $\text{macs}_1(a, b) > 0$, a contradiction.
(ii) Assume true for \( m \). If \( \text{macs}_1(a, b) = m + 1 > 0 \), it follows that \( a \oplus b = (a - 1) \oplus (b - 1) \), so Lemma 4 implies \( \text{macs}(a, b) = \text{macs}(a - 1, b - 1) + 1 \). But \( \text{macs}_1(a - 1, b - 1) = m \), so according to the induction hypothesis \( \text{macs}(a - 1, b - 1) = m \), hence \( \text{macs}(a, b) = m + 1 \).

Corollary 1 gives us an algorithm for computing \( \text{macs}_1(a, b) \) in linear time, namely, by computing \( \text{macs}(a, b) \). We will see how to compute \( \text{macs}_k(a, b) \) for any fixed \( k \) in time linear in the input size \( \log(1 + \max(a, b)) \), where \( k \) is considered constant for the time analysis.

**LEMMA 5.** If \( k = k_1 2^j \) then \( \text{macs}_k(a, b) = \text{macs}_k(\lfloor a/2^j \rfloor, \lfloor b/2^j \rfloor) \).

**Proof.** By definition of \( \text{macs}_n(a, b) \), it suffices to show

\[
\left\lfloor \frac{a}{2^j} \right\rfloor \oplus \left\lfloor \frac{b}{2^j} \right\rfloor = \left( \left\lfloor \frac{a}{2^j} \right\rfloor - ik_1 \right) \oplus \left( \left\lfloor \frac{b}{2^j} \right\rfloor - ik_1 \right)
\]

if and only if

\[
a \oplus b = (a - ik) \oplus (b - ik) = (a - ik_1 2^j) \oplus (b - ik_1 2^j).
\]

Indeed, since \( ik_1 2^j \) ends with at least \( j \) 0’s, it follows that \( (a - ik_1 2^j) \oplus (b - ik_1 2^j) \) has the same \( j \) bits suffix as \( a \oplus b \), and the two numbers are equal if and only if their prefixes are the same, i.e., if and only if \( \left\lfloor a/2^j \right\rfloor \oplus \left\lfloor b/2^j \right\rfloor = \left\lfloor a/2^j \right\rfloor \oplus \left\lfloor b/2^j \right\rfloor \).

The last lemma and Corollary 1 give us a complete formula for \( \text{macs}_k(a, b) \) when \( k \) is a power of 2: \( \text{macs}_k(a, b) = \text{macs}(\lfloor a/2^j \rfloor, \lfloor b/2^j \rfloor) \). In other words, \( \text{macs}_2(a, b) \) is the value of the common right bits of \( a \) and \( b \) after truncating their rightmost \( j \) bits.

All we have to do is to show how to compute \( \text{macs}_k(a, b) \) for an odd \( k > 1 \). We do not give a closed formula as in the case \( k = 2^j \), but the following two lemmas show how to compute it efficiently. In these lemmas we denote by \( t \) the number of common right bits of \( a \) and \( b \) for \( a \neq b \), i.e., \( t = \eta(a \oplus b) \). If \( a = b \), we can define \( t \) to be infinity and it will be consistent with the results below, but for simplicity we deal with the case \( a = b \) separately.

**LEMMA 6.** For every odd \( k \geq 1 \) if \( a \neq b \), then \( \text{macs}_k(a, b) < 2^t \).

**Proof.** Since \( a \neq b \) and \( k \) is odd, exactly one of \( a' \) and \( b' \) (the zero) absorbs a borrow from the \( t + 1 \) bit in subtracting \( 2^t k \) (cf. Lemma 3 above). Hence, \( (a - 2^t k) \oplus (b - 2^t k) \) and \( a \oplus b \) differ in bit position \( t + 1 \).
Lemma 7. For every $k \geq 1$, if $a = b$ or else $k \leq 2^t$, then

$$\text{macs}_k(a, b) = \left\lfloor \frac{\text{macs}(a, b)}{k} \right\rfloor.$$\

Proof. Show first $\text{macs}_k(a, b) \geq \lfloor \text{macs}(a, b)/k \rfloor$. Write $\text{macs}(a, b) = qk + k'$, where $0 \leq k' < k$. By Corollary 1, $a \oplus b = (a - i) \oplus (b - i)$ for every $i \leq qk + k'$, in particular for $i = k, 2k, ..., qk$. So, $\text{macs}_k(a, b) \geq q = \lfloor \text{macs}(a, b)/k \rfloor$.

If $a = b$, then $\text{macs}_k(a, b) = q$, since $a - qk \geq 0$, $a - (q + 1)k < 0$; so assume $a \neq b$. Since $a' \neq b'$, it follows that $a \oplus b \neq (a - i) \oplus (b - i)$ for any $i \in [qk + k' + 1, qk + k' + 2')$ such that $a - i \geq 0$ and $b - i \geq 0$, in particular for $i = (q + 1)k$, which is in this interval since $k \leq 2^t$. So, $\text{macs}_k(a, b) < q + 1$, and we have $\text{macs}_k(a, b) = q = \lfloor \text{macs}(a, b)/k \rfloor$. 

Algorithm $\text{macs}_k$. We construct a linear algorithm for computing $\text{macs}_k(a, b)$.

1. Write $k = 2^t k_1$, where $k_1$ is odd.
2. $a \leftarrow \lfloor a/2^t \rfloor$, $b \leftarrow \lfloor b/2^t \rfloor$, $k \leftarrow k_1$ (right-shifting).
3. If $a = b$ then return $\lfloor a/k \rfloor$.
4. $t \leftarrow$ the number of common right bits of $a$ and $b$.
5. If $k < 2^t$ then return $\lfloor \text{macs}(a, b)/k \rfloor$.
6. $i \leftarrow 1$.
7. While $a \oplus b = (a - ik) \oplus (b - ik)$ do
   7.1 $i \leftarrow i + 1$.
8. return $(i - 1)$.

The correctness of the algorithm stems from the lemmas: $\text{macs}_k(a, b)$ is invariant under the operation of step 2 by Lemma 5, steps 3–5 are valid by Lemma 7, and steps 6–8 are the naive way to calculate $\text{macs}_k(a, b)$. All we have to do for showing the linearity is to bound the number of times step 7 can be done. But since we reach this step only when $2^t < k$ and $a \neq b$, and since $k$ is odd, Lemma 6 implies $\text{macs}_k(a, b) < 2^t < k$. So we have an upper bound of $k$ which is a constant.

The complexity of the last algorithm is $O(k \log(1 + \max(a, b)))$, which is linear in the input size, but exponential in the size $\log k$ of the constant $k$. However, since $k$ may be large in the subsequent application, it is of interest to find an algorithm which is polynomial in $\log(\max(a, b, k))$. A recursive algorithm which is square in $\log(\max(a, b, k))$ is given below.
**Cartesian Coordinates, Diagonal Coordinates, and Bit Coordinates**

For constructing the recursive algorithm, we need a nonstandard coordinate system. Various coordinate systems can be seen in the grid of Nim depicted in Fig. 2. Every entry in the grid has a column number \( a \) and a row number \( b \). The numbers \( a \) and \( b \) are called the Cartesian coordinates of the grid point. For calculating \( \text{macs}_k(a, b) \), it is useful to look at all the grid points with the same value. In Fig. 2 we took Nim-value 5 as an example and boldfaced all its occurrences. We can see that 5 appears in diagonals. We distinguish between long diagonals (\( l \)-diagonals for short), which are all the grid points \( (a, b) \) with the same difference \( l = a - b \), and short diagonals (\( s \)-diagonals for short), which are all the grid points \( (a, b) \) with the same sum \( s = a + b \).

The numbers \( l \) and \( s \) are called the diagonal coordinates of the grid point. In our example, we see that the number 5 appears only in the four \( l \)-diagonals \(-5, -3, 3, 5\). (There are \( 4 = 2^m \) such diagonals where \( m = 2 \), since 5 has \( m = 2 \) 1-bits in its binary encoding.) On the other hand, we see the number 5 in the \( s \)-diagonals \( 9, 21, 25, 37, 41, \ldots \) (we give the law in Fig. 2. The grid of Nim.)

### Fig. 2. The grid of Nim.

<table>
<thead>
<tr>
<th>( a )</th>
<th>( b )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
</tr>
</tbody>
</table>

...
Moreover, the number 5 appears in all and only all grid points with coordinates \((l, s)\) from the above lists.

The importance of the diagonal coordinates is seen in calculating \(\text{macs}_k(a, b)\). The subtraction of \(k\) from both \(a\) and \(b\) is equivalent to "jumping" from the grid point \((a, b)\) along an \(l\)-diagonal, where the \(s\)-coordinate decreases by \(2k\); in the diagonal coordinates system: \((l, s) \rightarrow (l, s - 2k)\).

The following lemma gives a formula for the distance between a grid point \((a, b)\) and the grid point with the same \(l\)-coordinate and with \(s\)-coordinate \(s = a \oplus b\). (Since \(a \oplus b < a + b\), it follows that the \(s\)-diagonal \(a \oplus b\) is the minimal \(s\)-diagonal in which \(a \oplus b\) appears.)

**Notation.** For \(x, y \in \mathcal{Z}^0\), define their and by \(x \& y = \sum_{i \geq 0} x_i y_i 2^i\).

**Lemma 8.** For every \(a, b \in \mathcal{Z}^0\), we have \((a + b - (a \oplus b))/2 = a \& b\).

**Proof.** Write \(c = a + b - (a \oplus b)\), \(d = a \& b\). Examining the bits of \(c\) shows that \(c^i = 1\) \((i > 0)\) if and only if there is a carry during addition and subtraction from digital position \(i - 1\), and this happens if and only if \(a^{i-1} = b^{i-1} = 1\) if and only if \(d^i = 1\). Since \(c^0 = 0\), it follows that \(c = 2d\).

**Theorem 3.** For every \(a, b \geq 0\), we have \((a \oplus b) \& (a \& b) = 0\). Conversely, for every \(x, y \geq 0\), if \(x \& y = 0\) then there are \(2^m\) pairs \(\{(a_1 b_1), \ldots, (a_{2^m} b_{2^m})\}\) with \(a_i \oplus b_i = x\) and \(a_i \& b_i = y\) \((1 \leq i \leq 2^m)\), such that \(a_u - b_u \neq a_v - b_v\) for \(u \neq v\), where \(m\) is the number of 1's in the binary encoding of \(x\). Moreover, if we take \(x\) and \(y'\) with \(x \& y' = 0\), then the sequence of pairs \(\{(a'_1, b'_1), \ldots, (a'_{2^m}, b'_{2^m})\}\) satisfies \(\{a'_i - b'_i : 1 \leq i \leq 2^m\} = \{a_i - b_i : 1 \leq i \leq 2^m\}\). (In other words, the points \((a_i, b_i)\) and \((a'_i, b'_i)\) are on the same \(l\)-diagonal.)

**Proof.** Clearly \((a \oplus b) \& (a \& b) = 0\).

Conversely, assume \(x \& y = 0\). Define \(a\) and \(b\) as follows: for every \(i \geq 0\), if \(y^i = 1\) then \(a^i = b^i = 1\); if \(x^i = 1\) (and so \(y^i = 0\)) then either \(a^i = 1, b^i = 0\) or \(a^i = 0, b^i = 1\); if \(x^i = y^i = 0\) then \(a^i = b^i = 0\). We got in this way \(2^m\) pairs \((a, b)\), where \(m\) is the number of 1’s in the binary encoding of \(x\), all of them satisfy \(a_i \oplus b_i = x\) and \(a_i \& b_i = y\), \(i = 1, \ldots, 2^m\). Let \((a_u, b_u)\) and \((a_v, b_v)\) be two pairs constructed from two different choices of \(a^i, b^i\) when \(x^i = 1\). Let \(j\) be the maximal digital position with \(a^j \neq b^j\). Then \(a_u + b_v\) and \(a_v + b_u\) differ in their digital position \(j + 1\), so \(a_u - b_u \neq a_v - b_v\). Moreover, once we chose \(a', b'\) for those \(i\)'s in which \(x^i = 1\), the difference \(a - b\) is independent of \(y\). Indeed, \(a' = b' = 1\) for every \(y' = 1\), and in the subtraction they are cancelled. Hence, only \(x\) determines the \(l\)-diagonals in which the points appear.
Lemma 8 and Theorem 3 give us a formula for all the $s$-diagonals $s$ in which the value $x$ appears, namely $s = 2y + x$, where $y \& x = 0$. In our example $x = 5$ (101 in binary), the possible $y$ are 0, 2, 8, 10, 16, 18, ... so the $s$'s are 5, 9, 21, 25, 37, 41, ... 

Corollary 2. For every $a, b, a', b' \geq 0$, if $a + b = a' + b'$ and $a \oplus b = a' \oplus b' = x$, then for every $k > 0$, $\text{macs}_k(a, b) = \text{macs}_k(a', b')$.

Proof. It suffices to show that $(a - j) \oplus (b - j) = x$ if and only if $(a' - j) \oplus (b' - j) = x$ for every $j$ satisfying $0 \leq j \leq \min\{a, b, a', b'\}$. Indeed, according to Theorem 3 and Lemma 8, $(a - j) \oplus (b - j) = x$ if and only if $((a + b - x - 2j)/2) \& x = 0$ if and only if $((a' + b' - x - 2j)/2) \& x = 0$ if and only if $(a' - j) \oplus (b' - j) = x$.

Let us do one more step and concentrate only on $x = a \oplus b$ and $y = a \& b$. The numbers $x$ and $y$ are called the bit coordinates of the grid point. Unlike the above coordinates, bit coordinates do not define a unique grid point, but a group of grid points, namely all the grid points with common $s$-coordinate $s = 2y + x$ and the same value $x$. Corollary 2 justifies use of bit coordinates, since $\text{macs}_k(a, b)$ depends only upon them. Thus we can redefine $\text{macs}_k(a, b)$ with the help of $x$ and $y$, where $x = a \oplus b$ and $y = a \& b$:

$$\text{macs}_k(x, y) = \max\{d : (y - ik) \& x = 0 \text{ for every } i \leq d\}.$$ 

Recall that $\text{and}$ is defined only on nonnegative integers.

Notation. From now on until the end of the section $r = \lceil \log_2(x + 1) \rceil$.

If we examine the new definition of $\text{macs}_k(x, y)$, we see that for $j = \text{macs}_k(x, y) + 1$, there are two possibilities: either $y - jk < 0$ or else $(y - jk) \& x \neq 0$. But since $x' = 0$ for every $i \geq r$, it follows that $(y - jk) \& x \neq 0$ if and only if $(y - jk)(2^r) \& x \neq 0$. This motivates the definition of the cyclic $\text{macs}$, or $\text{cmacs}$ for short.

Definition. Let $x, y \geq 0$ be bit coordinates. For every $k > 0$ not divisible by $2^r$, define

$$\text{cmacs}_k(x, y) = \max\{d : (y - ik)(2^r) \& x = 0 \text{ for every } i \leq d\}.$$ 

It is not clear that this maximum exists, since it may be infinity (and this is what happens when $2^r \mid k$). However, Theorem 4 below shows how to compute it, and incidentally shows that it is well defined.

The definition of $\text{cmacs}$ implies that $\text{cmacs}_k(x, y) = \text{cmacs}_{k(2^r)}(x, y(2^r))$. So it suffices to compute $\text{cmacs}$ only for $k, y < 2^r$. 


Notation. If \( x > 0 \), define the \( r \)-complement of \( x \) by \( \hat{x} = \sum_{i=0}^{r-1} (1 - x^i)2^i \).

Note that the leading 0's are not complemented.

**Theorem 4.** Let \( x > 0 \) and \( 0 \leq y < 2^r \) be bit coordinates, and let \( 0 < k < 2^r \). Define \( s < r \) to be the minimal digital position such that \( x^i = 1 \) for every \( i \in [s, r-1] \). If \( k \leq 2^{r-1} \) then \( \text{cmacs}_k(x, y) = \text{macs}_k(x(2^s), y) \). If \( 2^{r-1} < k < 2^r \) then \( \text{cmacs}_k(x, y) = \text{cmacs}_{2^r-k}(x, \hat{x} - y) = \text{macs}_{2^r-k}(x(2^s), \hat{x} - y) \).

**Proof.** Clearly \( y < 2^r \), so \( (y-ik) \& x = 0 \) if and only if \( (y-ik) \& x(2^r) = 0 \). Hence \( \text{macs}_k(x, y) = \text{macs}_k(x(s), y) \). Assume \( k \leq 2^{r-1} \), then \( \text{cmacs}_k(x, y) \geq \text{macs}_k(x, y) \) because \( (y-ik) \& x = 0 \Rightarrow (y-ik)(2^r) \& x = 0 \).

On the other hand, if \( \text{cmacs}_k(x, y) > \text{macs}_k(x, y) \), then for \( i = \text{macs}_k(x, y) \), we must have \( y-ik \geq 0 \) and \( -2^{r-1} \leq -k \leq y-(i+1)k < 0 \). But then \( 2^{r-1} \leq (y-(i+1)k)(2^r) < 2^r \), so \( (y-(i+1)k)(2^r) \& x \neq 0 \), a contradiction.

Assume that \( 2^{r-1} < k < 2^r \). We will show that for every \( i \geq 0 \), \( (y-ik)(2^r) \& x = 0 \) if and only if \( (\hat{x} - y - i(2^r-k))(2^r) \& x = 0 \). Indeed,

\[
(y-ik)(2^r) \& x = 0 \Rightarrow (\hat{x} - (y-ik)(2^r)) \& x = 0 \\
\Rightarrow (\hat{x} - y - i(2^r-k))(2^r) \& x = 0; 
\]

conversely,

\[
(\hat{x} - y - i(2^r-k))(2^r) \& x = 0 \Rightarrow (\hat{x} - (\hat{x} - y - i(2^r-k))(2^r)) \& x = 0 \\
\Rightarrow (y-ik)(2^r) \& x = 0. 
\]

The last equality in the theorem follows from the first part of the theorem since \( 2^{r-k} < 2^{r-1} \).

Theorem 4 enables us to express \( \text{cmacs} \) in term of \( \text{macs} \); the following lemma expresses \( \text{macs} \) in term of \( \text{cmacs} \). Thus we can compute \( \text{macs}_k(x, y) \) recursively.

**Lemma 9.** For all bit coordinates \( x, y \geq 0 \) and \( k > 0 \), if \( 2^r \mid k \) then \( \text{macs}_k(x, y) = \lfloor y/k \rfloor \); otherwise \( \text{macs}_k(x, y) = \min \{ \lfloor y/k \rfloor, \text{cmacs}_{2^r}(x, y) \} \).

**Proof.** If \( 2^r \mid k \) then \( (y-ik) \& x = 0 \) for every \( i \geq 0 \) for which the and is defined. Thus, \( \text{macs}_k(x, y) = \lfloor y/k \rfloor \).

Otherwise, we have to prove three assertions:

(i) \( \text{macs}_k(x, y) \leq \lfloor y/k \rfloor \).

(ii) \( \text{macs}_k(x, y) \leq \text{cmacs}_k(x, y) \).

(iii) \( \text{macs}_k(x, y) \geq \min \{ \lfloor y/k \rfloor, \text{cmacs}_{2^r}(x, y(2^r)) \} \).
The first two assertions are clear; for the third one it suffices to show that for every \( i \geq 0 \), if \( i \leq \lfloor y/k \rfloor \) and \( i \leq \text{macs}_k(x, y) \), then \((y - ik) \& x = 0\). Indeed, \( y - ik \geq 0 \), so the expression is well defined. Since \( x^j = 0 \) for every \( j \geq r \), it follows that \((y - ik)(2^r) \& x = 0 \Rightarrow (y - ik) \& x = 0\). \]

**Algorithm rmacs\(_k\) (a recursive form).** The algorithm computes the function \( \text{macs}_k(x, y) \) when \((x, y)\) are the bit coordinates of a grid point.

\[
\begin{align*}
\text{rmacs}_k(x, y) = \langle n & \rangle \\
1. & r \leftarrow \lceil \log_2(x + 1) \rceil \\
2. & \text{If } 2^r | k \text{ then return } (\lfloor y/k \rfloor) \\
3. & y \leftarrow y(2^r) \\
4. & k \leftarrow k(2^r) \\
5. & \text{If } k > 2^{r-1} \text{ then} \\
5.1 & k \leftarrow 2^r - k \\
5.2 & y \leftarrow x - y \\
6. & \text{Let } s \leq r \text{ be the minimal digital position such that } x^i = 1 \text{ for every } i \in [s, r - 1] \\
7. & x \leftarrow x(2^s) \\
8. & \text{return } \langle \lfloor y/k \rfloor, \text{rmacs}_k(x, y) \rangle.
\end{align*}
\]

The correctness of the algorithm stems from Lemma 9 and Theorem 4 when we skip the intermediate step of using cmacs and express macs with \( \text{macs}_k \). The complexity of one iteration is linear in \( \log(\max(a, b, k)) \), and the number of iterations is bounded by \( \lceil \log_2(x + 1)/2 \rceil \). (The worst case is when the binary encoding of \( x \) is 101010....)

A good idea for efficiency is to combine parts of both algorithms. Thus, we can first make \( k \) odd by right-shifting. Right-shifting of \( a \) and \( b \) by \( j \) positions is equivalent to right-shifting of \( x \) and \( y \) by \( j \) positions. If \( k \) is odd at the beginning, it remains odd during the recursion, so it is better to do the shifting outside the recursion only once. If \( k \) is odd, the condition \( 2^r \mid k \) is equivalent to \( x = 0 \). We can end the algorithm when \( k < 2^t \), where \( t = \eta(x) (x \neq 0) \). In this case we have to redefine macs in terms of bit coordinates, namely, \( \text{macs}(a, b) = y(2^t) \). The naive part of algorithm \( \text{macs}_k \) (steps 6–8) also can be used: Recall that when \( k \) is odd, \( \text{macs}_k(x, y) < 2^t \). So we can compare \( 2^t \) with the number of times the recursion repeats and decide which algorithm to choose. Since \( t \) is not changed during the recursion, this part should also be outside the recursion. The last improvement we make is to ask at the beginning whether \( \lfloor y/k \rfloor = 0 \). If so, we can halt the recursion with the value 0. These ideas are realized in the following algorithm.
ALGORITHM omacs\(k\) (optimized version). An optimized algorithm for computing \(\text{macs}_k(x, y)\).

1. Write \(k = 2^j k_1\), where \(k_1\) is odd.
2. \(x \leftarrow \lfloor x/2^j \rfloor\), \(y \leftarrow \lfloor y/2^j \rfloor\), \(k \leftarrow k_1\) (right-shiting).
3. If \(x = 0\) then return \((\lfloor y/k \rfloor)\).
4. \(t \leftarrow \eta(x)\).
5. If \(k \leq 2^t\) then return \((\lfloor y(2^t)/k \rfloor)\).
6. compute the cardinality \(c = \# \{i \geq 0 : x^i = 1\text{ and if } i > 0 \text{ then } x^{i-1} = 0\}\).
7. If \(2^t \leq c\) then
   7.1. \(i \leftarrow 1\).
   7.2. While \((y - ik) \& x = 0\) do
       7.2.1. \(i \leftarrow i + 1\).
   7.3. return \((i - 1)\).
8. return \((\text{rec-macs}_k(x, y))\).

\text{rec-macs}_k(x, y)

1. \(d \leftarrow \lfloor y/k \rfloor\).
2. If \(x = 0\) or \(d = 0\), then return \((d)\).
3. \(r \leftarrow \lceil \log_2(x + 1) \rceil\).
4. \(y \leftarrow y(2^r)\).
5. \(k \leftarrow k(2^r)\).
6. If \(k \leq 2^t\) then return \((\lfloor y(2^t)/k \rfloor)\).
7. If \(k > 2^{t-1}\) then
   7.1. \(k \leftarrow 2^t - k\).
   7.2. \(y \leftarrow x - y\).
8. Let \(s \leq r\) be the minimal digital position such that \(x^i = 1\) for every \(i \in [s, r - 1]\).
9. \(x \leftarrow x(2^t)\).
10. return \((\min\{d, \text{rec-macs}_k(x, y)\})\).

6. DOUBLE CYCLIC NIMHOFF

Double cyclic Nimhoff is the class of 2-pile Nimhoff\((R)\) when \(R \subseteq (\mathcal{S}^+)^2\) is a relation of the type \(R(x, y)\) if \(x + y < h\) or \(x + y = 2h\), where \(h > 1\) is a
given integer. (From this section on, \((a, b)\) denotes a game position and \((x, y)\) a move from that position.) The name double cyclic comes from the structure of the Sprague-Grundy values' grid. Recall that for every \(w \geq 0\), 
\[
\bar{w} = \lfloor w/h \rfloor.
\]

**Theorem 5.** The \(g\)-function of a position \((a, b)\) of double cyclic Nimhoff is

\[
g(a, b) = h(\bar{a} \oplus \bar{b}) + (a + b - \text{macs}(\bar{a}, \bar{b}))(h).
\]

**Proof.** Denote by \(G(a, b)\) the right-hand side of formula (6).

(A) Let \((a', b')\) be an option of \((a, b)\), and assume that \(G(a, b) = G(a', b')\), where \(x = a - a', y = b - b'\). Then by (6),

\[
\bar{a} \oplus \bar{b} = \bar{a}' \oplus \bar{b}',
\]

\[
a + b - \text{macs}(\bar{a}, \bar{b}) \equiv a' + b' - \text{macs}(\bar{a}', \bar{b}') \pmod{h}.
\]

Since \((a', b')\) is an option of \((a, b)\), there are three possibilities:

(I) \(a' < a, b' = b\). (The symmetric case \(a' = a, b' < b\) is similar.) If \(\bar{a}' < \bar{a}\) then we have a contradiction to (7). If \(\bar{a}' = \bar{a}\) then we have a contradiction to (8).

(II) \(x + y < h\) with \(x, y > 0\). Then \(a' < a, b' < b\). We consider three subcases:

1. \(\bar{a}' = \bar{a}, \bar{b}' = \bar{b}\). This contradicts (8), since \(0 < a + b - (a' + b') < h\).

2. \(\bar{a}' = \bar{a} - 1, \bar{b}' = \bar{b} - 1\). From (7) and Lemma 4 we obtain

\[
\text{macs}(\bar{a}'', \bar{b}'') = \text{macs}(\bar{a}, \bar{b}) - 1.
\]

Since \(x + y > 1\), it follows that

\[
a + b - \text{macs}(\bar{a}, \bar{b}) - (a' + b' - \text{macs}(\bar{a}', \bar{b}')) = x + y - 1 \not\equiv 0 \pmod{h},
\]

contradicting (8).

3. \(\bar{a}' = \bar{a}, \bar{b}' = \bar{b} - 1\) (or \(\bar{a}' = \bar{a} - 1, \bar{b}' = \bar{b}\)) contradicts (7).

(III) \(x + y = 2h\) with \(x, y > 0\). If \(\bar{a}' = \bar{a} - 1, \bar{b}' = \bar{b} - 1\), then again, from (7) and Lemma 4 we obtain (9) and (10). Otherwise \((\bar{a}' = \bar{a} - 2, \bar{b}' = \bar{b} - 1\) or \(\bar{a}' < \bar{a}, \bar{b}' = \bar{b}\) or the symmetric cases), all the possibilities lead to a contradiction to (7).
(B) Write $G = G(a, b)$, and let $G' < G$. To show that there is an option $(a', b')$ of $(a, b)$ such that $G(a', b') = G'$, we consider two cases:

(I) $G' < G$. Since $\overline{G} = \overline{a} \oplus \overline{b}$, without loss of generality there is $\overline{A} < \overline{a}$ such that $\overline{A} \oplus \overline{b} = G'$. Take $a' = h \cdot \overline{A} + (G' - b + \text{macs}(\overline{A}, \overline{b}))(h)$ and $b' = b$. Clearly $a' < a$, so $(a', b')$ is indeed an option of $(a, b)$. Moreover, since $\overline{a'} = \overline{A}$ and $\overline{b'} = \overline{b}$,

$$\overline{G(a', b')} = \overline{a'} \oplus \overline{b'} = \overline{A} \oplus \overline{b} = G', \tag{11}$$

$$G(a', b')(h) = (a' + b' - \text{macs}(\overline{a'}, \overline{b'}))(h)$$

$$= (a'(h) + b(h) - \text{macs}(\overline{A}, \overline{b}))(h)$$

$$= (G'(h) - b(h) + \text{macs}(\overline{A}, \overline{b}))(h) + b(h) - \text{macs}(\overline{A}, \overline{b}))(h)$$

$$= G'(h). \tag{12}$$

From (11) and (12) we conclude $G(a', b') = G'$.

(II) $G' = G$.

1. $G - G' \leq a(h)$. Take $a' = a - (G - G')$ and $b' = b$. The last assumption implies

$$\overline{G(a', b')} = \overline{a'} \oplus \overline{b'} = \overline{a} \oplus \overline{b} = G = G', \tag{13}$$

$$G(a', b')(h) = (a'(h) + b'(h) - \text{macs}(\overline{a'}, \overline{b'}))(h)$$

$$= (a(h) - G(h) + G'(h) + b(h) - \text{macs}(\overline{a}, \overline{b}))(h)$$

$$= ((a(h) + b(h) - \text{macs}(\overline{a}, \overline{b}))(h) - G(h) + G'(h))(h)$$

$$= G'(h).$$

2. $a(h) < G - G' \leq a(h) + b(h)$. Take $a' = a - a(h)$ and $b' = b - (G - G' - a(h)) \geq b - b(h)$. Since $a - a' + b - b' = a(h) + G - G' - a(h) < h$, it follows that $(a', b')$ is an option of $(a, b)$. Moreover, we have $\overline{a'} = \overline{a}$ and $\overline{b'} = \overline{b}$, so (13) holds and

$$G(a', b')(h) = (a'(h) + b'(h) - \text{macs}(\overline{a'}, \overline{b'}))(h)$$

$$= (0 + b(h) - G(h) + G'(h) + a(h) - \text{macs}(\overline{a}, \overline{b}))(h)$$

$$= ((a(h) + b(h) - \text{macs}(\overline{a}, \overline{b}))(h) - G(h) + G'(h))(h)$$

$$= G'(h).$$

3. $G - G' > a(h) + b(h)$. Then $\text{macs}(\overline{a}, \overline{b}) \not\equiv 0 \pmod{h}$ because otherwise $a(h) + b(h) \geq (a \mid b)(h) = G(h) \geq G - G'$.
3.1. \( G - G' < h - 1 \). Take \( a' = a - a(h) \) and \( b' = b - (G - G' - a(h)) < b - b(h) \). Note that \( b' < 0 \Rightarrow b < G - G' - a(h) < h - 1 - a(h) < h \), hence \( a' \) or \( b' \) negative would imply \( \text{macs}(\bar{a}, \bar{b}) = 0 \), a contradiction. Thus \( 0 < a' < a \), \( 0 < b' < b \). Moreover,

\[
a - a' + b - b' = a(h) + 1 + G - G' - a(h) = G - G' + 1 < h.
\]

Therefore \((a', b')\) is indeed an option of \((a, b)\). It is easy to see that \( \bar{a} = \bar{a} - 1 \) and \( \bar{b} = \bar{b} - 1 \). Thus,

\[
\overline{G(a', b')} = \overline{a'} \oplus \overline{b'} = (\bar{a} - 1) \oplus (\bar{b} - 1) = \bar{a} \oplus \bar{b} = \overline{G} = \overline{G'}, \tag{14}
\]

where the third equality comes from the fact that \( \text{macs}(\bar{a}, \bar{b}) > 0 \) and Lemma 4.

\[
G(a', b')(h) = (a' + b' - \text{macs}(\overline{a'}, \overline{b'}))(h)
= (a - a(h) - 1 + b - (G - G' - a(h)) - \text{macs}(\bar{a} - 1, \bar{b} - 1))(h)
= (a + b - 1 + G' - G - \text{macs}(\bar{a}, \bar{b}) + 1)(h)
= ((a + b - \text{macs}(\bar{a}, \bar{b}))(h) + G' - G)(h)
= G'(h).
\]

3.2. \( G - G' = h - 1 \). Then \( G(h) = h - 1 \) and \( G'(h) = 0 \). Take \( a' = a - h \) and \( b' = b - h \). As in the previous subcase we conclude that \( a' \) and \( b' \) are nonnegative. Moreover, \( \bar{a} = \bar{a} - 1 \) and \( \bar{b} = \bar{b} - 1 \). Hence \( (14) \) holds, and

\[
G(a', b')(h) = (a' + b' - \text{macs}(\overline{a'}, \overline{b'}))(h)
= (a - h + b - h - \text{macs}(\bar{a}, \bar{b}) + 1)(h)
= ((a + b - \text{macs}(\bar{a}, \bar{b}))(h) + 1)(h)
= (G(h) + 1)(h) = 0 = G'(h). \]

Redundant comoves and nucleus. Like in cyclic Nimhoff, all the comoves \((x, y)\) of the type \( x + y < h \) are indeed needed. The second type of comoves \((x, y)\) with \( x + y = 2h \) is used by the winner only to remove an equal number \( h \) of tokens from every pile, the rest of the comoves are redundant. So the nucleus consists of all \((x, y)\) with \( x + y < h \) or \( x = y = h \).
7. Even Balanced Nimhoff

Given two piles of tokens, let \( R \subseteq (2^+)^2 \) be a relation of the type \( R(x, y) \) if \( x + y = 4l \), where \( l \geq 1 \) is a given integer. The name even balanced Nimhoff comes from the fact that the nucleus of the game is when \( x = y = 2l \).

**Lemma 10.** If \( a \oplus b = c \oplus d \) and \( c + d = a + b - 2l \) then \( \text{macs}_l(c, d) = \text{macs}_l(a, b) - 1 \).

**Proof.** Let \((x, y)\) be the bit coordinates of \((a, b)\). By Lemma 8, the bit coordinates of \((c, d)\) are \((x, y - l)\). Hence, the definition of \( \text{macs}_k(x, y) \) (the bit coordinates version) implies \( \text{macs}_l(x, y - l) = \text{macs}_l(x, y) - 1 \).

**Notation.** For every \( w \geq 0 \) let \( \bar{w} = \lfloor w/2 \rfloor \).

**Theorem 6.** The \( g \)-function of a position \((a, b)\) of even balanced Nimhoff is

\[
g(a, b) = \begin{cases} 
  a \oplus b & \text{if } \text{macs}_l(a, b) \text{ is even;} \\
  \ell(a \oplus b) & \text{if } \text{macs}_l(a, b) \text{ is odd.}
\end{cases}
\]

**Proof.** Denote by \( G(a, b) \) the right-hand side of the formula.

(A) Let \((a', b')\) be an option of \((a, b)\). There are two possibilities:

1. \( a' < a, b' = b \). (The symmetric case \( a' = a, b' < b \) is similar.) If \( \bar{a}' = \bar{a} \) then \( a \oplus b \neq a' \oplus b \), so \( \ell(a \oplus b) \neq \ell(a' \oplus b) \), hence \( G(a', b) \neq G(a, b) \); else \( a \oplus b \) and \( a' \oplus b \) differ in at least one bit other than the rightmost bit, so in any case \( G(a', b) \neq G(a, b) \).

2. \( a' + b' = a + b - 4l \). Then it easily follows that \( |\bar{a}' + \bar{b}' - (\bar{a} + \bar{b} - 2l)| \leq 1 \). We consider two subcases:
   1. \( \bar{a}' + \bar{b}' = \bar{a} + \bar{b} \). Again \( a \oplus b \) and \( a' \oplus b' \) differ in a bit other than the rightmost one, so \( G(a, b) \neq G(a', b') \).
   2. \( \bar{a}' + \bar{b}' = \bar{a} + \bar{b} - 2l \). Then the numbers \( \bar{a}' + \bar{b}' \) and \( \bar{a} + \bar{b} - 2l \) have the same parity, so we have \( \bar{a}' + \bar{b}' = \bar{a} + \bar{b} - 2l \). According to Lemma 10, \( \text{macs}_l(\bar{a}', \bar{b}') \equiv \text{macs}_l(\bar{a}, \bar{b}) \pmod{2} \), but \( a \oplus b \equiv a' \oplus b' \pmod{2} \), so \( G(a, b) \) and \( G(a', b') \) differ in their rightmost position.

(B) Write \( G = G(a, b) \), and let \( G' < G \). To show that there is an option \((a', b')\) of \((a, b)\) such that \( G(a', b') = G' \), we consider two cases:

1. \( G' < G \). Without loss of generality, there is \( A < a \) such that \( A \oplus b = G' \). If \( G(A, b) = G' \) then take \( a' = A \) and \( b' = b \) and we are done. Otherwise, it is the case that \( G(A, b) = \ell(G') \), so take
a' = i(A) and b' = b. Since a' = A, we obtain G(a', b) = i(G(A, b)) = i(i(G')) = G'. Moreover, we have a' < a, because otherwise a' = a and G = G(a, b) = G(a', b) = G'.

(II) \( \overline{G} = \overline{G}. \) Then G' is even and \( G = G' + 1. \) We consider two subcases:

1. \( \text{macs}_i(\bar{a}, \bar{b}) \) is even. It follows that \( a \) and \( b \) do not have the same parity. If, say, \( a \) is odd, take \( a' = a - 1 \) and \( b' = b. \)

2. \( \text{macs}_i(\bar{a}, \bar{b}) \) is odd. By definition of \( \text{macs}_i(a, b) \), we obtain \( \bar{a} \oplus \bar{b} = (\bar{a} - \bar{l}) \oplus (\bar{b} - \bar{l}) \) and \( \text{macs}_i(\bar{a} - \bar{l}, \bar{b} - \bar{l}) = \text{macs}_i(\bar{a}, \bar{b}) - 1. \) Take \( a' = a - 2l \) and \( b' = b - 2l, \) so \( \bar{a'} = \bar{a} - \bar{l} \) and \( \bar{b'} = \bar{b} - \bar{l}, \) therefore \( \text{macs}_i(\bar{a'}, \bar{b'}) \) is even. Thus, we have \( G(a', b') = i(G(a, b)) = i(G) = G'. \)

Remarks. (a) Even balanced Nimhoff with \( l = 1 \) coincides with double cyclic Nimhoff with \( h = 2. \)

(b) When \( l \) is a power of 2, even balanced Nimhoff can be generalized to more than two piles, and the formula becomes \( k \)-Nim-sum.

Redundant comoves and nucleus. The nucleus of the game is when \( x = y = 2l; \) all the rest of the comoves are redundant. Since the g-value for even balanced Nimhoff is the same as for Nim except, possibly, in the lowest bit, it follows that every comove of the form \( (2x, xy) \), where \( (x, y) \) is in Nimdi, can be adjoined as a redundant comove (because \( G(a, b) \neq G(a', b') \)).

Generalizations?

It is easy to see that the formula for \( g(a, b) \) can be rewritten as

\[
g(a, b) = 2(\bar{a} \oplus \bar{b}) + (a + b - \text{macs}_i(\bar{a}, \bar{b}))(2).
\]

Familiar? Yes, the formula resembles the formula of double cyclic Nimhoff with the following changes: (i) double cyclic Nimhoff is the special case of even balanced Nimhoff when \( l = 1; \) (ii) on the other hand, even balanced Nimhoff is the special case of double cyclic Nimhoff when \( h = 2 \) (cf. remark (b)). This raises the question whether we can unify these classes under the formula \( g(a, b) = h(\bar{a} \oplus \bar{b}) + (a + b - \text{macs}_i(\bar{a}, \bar{b}))(\bar{w}), \) where \( \bar{w} = \lfloor w/h \rfloor. \) The natural candidate for \( R \) will be \( R(x, y) \) if \( x + y < h \) or \( x = y = lh. \) This is a good idea—but unfortunately it is not true. Examining examples of this class (e.g., \( h = 5, l = 3 \)) shows that the structure of the grid is destroyed under this modification.

Another unsuccessful direction to generalize double cyclic Nimhoff and even balanced Nimhoff is to more than two piles. The difficulty is hidden in the fact that the macs function is not well defined on more than two.
piles, for it depends on the order of the subtractions. For example, take \( n = 3 \) and \((a_1, a_2, a_3) = (1, 3, 3)\) with \(1 \oplus 3 \oplus 3 = 1\). We can subtract three times 1 from the second and third piles and remain with Nim-sum 1, so \(m(p, 3, 3) = 3\). On the other hand, if we subtract 1 from the first and second piles we would have \((0, 2, 3)\), and now it is impossible to subtract 1 from two piles without changing the Nim-sum, so \(m(p, 3, 3) = 1\). In this example the two values have the same parity \((1 \text{ and } 3)\). It is not surprising. In even balanced Nimhoff the formula depends only on the parity of \(m(p, 3, 3)\), and since this example deals with \(l = 1\) which is a power of 2, we know that it does have a generalization to \(n\) piles. Another example in which the parities are also not equal is \(m(p, 3, 7)\), which can be 2 or 1.

REFERENCES