university of groningen / faculty of science and engineering / mathematics and applied mathematics

The Nimbers

Bachelor's Project Mathematics

July 2022

Student: B. B. Tielman

First supervisor: Prof.dr. J. Top

Second assessor: Prof.dr. O. Lorscheid

Contents

1 Introduction

Nim is a really old game. It was already played in ancient China using nuts [6]. It is part of a collection of games called combinatorial games. These are games consisting of two players, where both can observe the state of the game and both take turns to move the game from one state to another.

There are many further restrictions to put on combinatorial games like ending in a finite number of steps, restricting that one cannot return to a previous state, or that the moves a player can make only depend on the state of the game itself and not on whose turn it is. Nim satisfies these restrictions. There are also many different variations of Nim, but in this thesis we will only talk about the game called "normal Nim."

It turns out that the game Nim gives rise to a rich structure under the right definitions, namely a tower of fields called $\mathbf{On}_2$. In this thesis we will show how the game Nim gives rise to this tower, link the tower to more generally understood notions in algebra, and end with showing that there are code which form vector spaces over the nimbers.

2 The Game Nim

Nim is a game played with a number of groups/heaps/piles, each consisting of some number of objects. Examples include stacks of coins being places next to each-other, rows of matchsticks, pockets of marbles and groups of nuts. We give some examples.

Figure 1: A game of Nim using coins.[1]

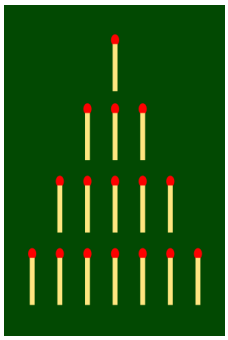

Figure 2: A game using matchsticks. The rows are the groups.[2]

[1] Image from https://www.pyth.eu/het-spel-nim
[2] Image from https://en.wikipedia.org/wiki/Nim

Two players each take turns to move. A move a player can make consist of removing any amount of objects in a single pile, after which it is the other player's turn. The game ends when the last player takes the last object, that player is then declared the winner.

3 Combinatorial Games

3.1 Definition of games

In combinatorial game theory, the following notation is used for a two player game. This notation is used by Conway in [2]. We have two different players, usually called player Left and Right. This does not mean player Left plays first and Right second, it is simply a distinction between players who can make different moves. A game x is described by the set of moves player Left can make, and the set of moves player Right can make, which need not be the same set. It is written like

$$x = \{l_1, l_2, l_3, \ldots \mid r_1, r_2, r_3, \ldots\},$$

where $\{l_1, l_2, l_3, \ldots\} = L$ and $\{r_1, r_2, r_3, \ldots\} = R$.

Notation. Sometimes we also write $x = \{L \mid R\}$. This is however not entirely correct since it identifies L and R with their respective elements.

We see x as an ordered pair of L and R for which $\{l_1, l_2, l_3, \ldots\} = L$ and similarly for R. The sets describe the moves player Left and player Right can make that lead to different game states, so formally $x = \{\{\{l_1, l_2, l_3, \ldots\}\}, \{\{l_1, l_2, l_3, \ldots\}, \{r_1, r_2, r_3, \ldots\}\}\}$. Each move, which is an element of either L or R, can then be seen as different state of our original game, or as a game in its own right. This means that the terms *move* and *game state* will be used somewhat interchangeably to describe the same element in L or R, although a move y is most often seen as the action of going from a game x to the element y, while a state y is seen as y itself.

A move in a game corresponds to picking an element of either L or R, depending on whose turn it is. We will give two small examples.

Example 1. Consider the following game. We have two lime (green) coins and two red coins. With this, we make a pile consisting of only one red coin and a different pile consisting of one red coin at the bottom, and two lime coins on top. Player Left may remove any number of lime coins that are on top of a pile and likewise, player Right may take any red coins that are on top. The player who cannot make a move loses.

Figure 3: The game (r)+(rll)

We introduce some notation to deal with this game. Let us describe any pile by small sequence of the colors of the coins form from bottom to top. Let us also denote separate piles by $+$. For now, this plus is noting but a symbol to us which means *the pile '' and the pile '' .* If we introduce the notation that $(r) + (rll)$ denotes the game with pile consisting of a single red coin and a pile of coins where the red is at the bottom and two limes are on top, then we can describe the game as follows.

The only move player Right can make if they were to start first, is to take away the single red coin, leaving the red-lime-lime pile unaltered. This means that $R = \{() + (rll)\}$. The move that Left

can make if they were to start first is to take away one or two coins from the pile of three. So they can make the game move to the states $(r)+(rl)$ and $(r)+(r)$, hence $L=\{(r)+(r),(r)+(rl)\}$. So we write

$$(r)+(rll)=\{(r)+(r),(r)+(rl)\,|\,()+(rll)\}.$$

Each of the elements of L and R are games themselves and should be analysed in the same way. For example, the game $()+(rll)$ is equal to $\{()+(rl),(r)+(r)\,|\quad\}$.

At the end of the analysis, we always come to the game $()+()$, which is equal to $\{\ |\ \}$. In other words, $L=\varnothing$ and $R=\varnothing$.

If we now remove the restriction of only making a move corresponding to one's color, then we get a game of Nim.

Example 2. We again have four coins, all blue. We again make two different piles, one consisting of a single coin and one consisting of three coins. Both players are allowed to remove any number of blue coins from a pile.

Figure 4: The game (b)+(bbb)

Since there is no difference in color anymore we only need to analyse what any player can do. A player can either remove the single coin from the pile containing one coin, or remove one, two or all coins from the pile containing three coins. This means that the move a player can make are $()+(bbb)$, $(b)+(bb)$, $(b)+(b)$ and $(b)+()$. This means that

$$(b)+(bbb)=\{()+(bbb),(b)+(bb),(b)+(b),(b)+()\,|\,()+(bbb),(b)+(bb),(b)+(b),(b)+()\}.$$

Note however that $\{\cdot|\cdot\}$ was shorthand notation for an ordered pair, so we could describe this game equally and more formally as

$$(b)+(bbb)=\{\{\{()+(bbb),(b)+(bb),(b)+(b),(b)+()\}\}\}.$$

Note that the parentheses occur trice.

We introduce some general notation for games. We denote a general element of L as x^L and of R as x^R. We sometimes also need to distinguish between the the set of moves player Left can make in the game x and the game y. These sets will be called L_x and L_y, and similarly R_x and R_y.

The game we will talk about in this thesis is the impartial game Nim. Impartial games have the property that the set of option for both players is identical and only depends on the game itself. This means that Left has the same set of options as Right, so $L=R$. This allows us to use change the notation for a game by writing x as just the set of moves one can make, as done in Lenstra's paper [8].

We cannot define games to contain just anything. Unlike in our examples, we actually need to construct games from the ground up to ensure that we do not end up with sets containing themselves.

Definition 3. A *game* x is pair of sets (L,R) where every element of L and R is a previously constructed games. The empty pair $(\varnothing,\varnothing)$ is a game.

Definition 4. An *impartial game* x is a set were every element is a previously constructed impartial games. The empty set $\varnothing$ is an impartial game.

This means that in order to make a game x, we should have already made all the previous games that make up x. It follows that every impartial game is a set whose elements consist of only sets. Likewise, a game is an ordered pair (L, R), whose elements are all ordered pairs.

The meaning behind this for the game Nim, is that we never see the same layout of the coins twice. For example, when we have single pile of seven coins and a player removes a coin, then the next player cannot place it back. (There is however a bit more nuance in how to visualize these sets as games.)

3.2 Addition of games

Addition of games is inspired by the intuition for how addition behaves on real numbers. When we have two **real numbers** x and y, we know that that for any number $x^L < x$ and $y^L < y$ that $x + y^L < x + y$ and $x^L + y < x + y$. And we have something similar if $x^R > x$ and $y^R > y$. This leads to the following definition of addition of games as found in [2]

Definition 5. The *sum of two games* x and y is given by the recursive formula

$$x + y := \bigcup_{\substack{x^L \in L_x \\ y^L \in L_y \\ x^R \in R_x \\ y^R \in R_y}} \{x^L + y,\, x + y^L \,|\, x^R + y,\, x + y^R\} \overset{\text{notation}}{=} \{x^L + y, x + y^L \,|\, x^R + y, x + y^R\}.$$

This means that the sum of x and y is the game where all the states Left can go to, are the states they can go to by making a move on x and leaving y unchanged, or making a move on y and leaving x unchanged, and analogously for Right.

For our impartial games, this definition gets compactified, since we do not distinguish between x^L and x^R anymore. We will use the notation x' for a general element of x. When we make these changes, the definition becomes the following, as seen in [8].

Definition 6. The *sum of two impartial games* x and y is given by the recursive formula

$$x + y := \bigcup_{\substack{x' \in x \\ y' \in y}} \{x' + y, x + y'\} \overset{\text{notation}}{=} \{x' + y,\, x + y'\}.$$

This also corresponds with our intuition of how a game of Nim with two piles should be played; your options are to either reduce the first pile x to some smaller pile x' and leave y unchanged or to reduce the second pile y to some smaller pile y' leaving x unchanged.

Note that this definition of addition is recursive. To find the sum of x and y we need to know the sums $x' + y$ and $x + y'$ for all elements in x and y, which in turn depend on their respective elements.

The definition is quite long to write down, which is why we introduce the shorthand notation. Whenever we write an $\{x'\}$, it is implied that we take the union over all x' in x of the elements that are described by it. In the case of addition, we see in the shorthand notation $\{x' + y, x + y'\}$. Note the appearance of an x' and a y', this means that we take a union over all elements in x and y of the different sums. What we get is the original full definition. This is analogous to Einstein notation where an index that appears in the top and bottom of an expression needs to be summed over. In the upcoming chapter we will proceed with the full notation, and later switch to the shorthand one. We will however stop with the distinction between games and impartial games, and only consider impartial games from here on out.

Notation. Whenever we write *game*, we mean *impartial game*.

We might worry about the definition of addition, since it is recursive. How do we know we will not end up with something that is not a set anymore? We propose the following lemma, which was proven by Lenstra [8].

Lemma 7. *Addition of games is well defined.*

Proof. We require some set theory to deal with this. We know by the axiom of regularity that there cannot be an infinite sequence of descending sets. In other words, we cannot have

$$x_1 \in x_2 \in x_3 \in \dots$$

This means that our games do not have infinite "depth," and that the recursion has to end at some point, meaning that the end result of our addition is again a set. □

We will first fully explore addition before we introduce multiplication. We need to look at the properties of addition, namely that addition of games is commutative and associative.

Lemma 8. *For any games x and y we have that addition is commutative. So $x + y = y + x$.*

Proof. The technique we use to prove this will be used a lot in this paper so it is important to go over it in depth the first time. When we compute $x + y$, we need the values of $x' + y$ and $x + y'$, but these calculations also depend on elements in their respective sets. Since we already showed that we do not have a infinite "depth" for our sets to go through, we must at some point reach the empty set for both x and y. This is clearly commutative since $\varnothing + \varnothing = \varnothing$ by our definition. We can thus conclude that sums of the form $\{\varnothing\} + \varnothing$ commute, which we can use to prove even more. This whole procedure is called backwards induction. When proving a property about games $P(x, y)$, one may assume that the property holds for all elements that preceded it, so $P(x', y)$, $P(x, y')$ and $P(x', y')$. We do not need a base case because in the end, we get a statement about the empty set which is vacuously true.

So with this all cleared, the proof of commutativity is quite straightforward, as done by Conway and Knuth in [2, 7]. We want to know whether $x+y = y+x$, but we can already assume $x'+y = y+x'$ and $x + y' = y' + x$ for all $x' \in x$ and all $y' \in y$, which leads us directly to the proof.

$$x + y = \bigcup_{\substack{x' \in x \\ y' \in y}} \{x' + y, x + y'\} = \bigcup_{\substack{x' \in x \\ y' \in y}} \{y + x', y' + x\} = y + x.$$

□

Lemma 9. *For any games x, y and z, we have that their sum is associative, so $(x + y) + z = x + (y + z)$.*

Proof. We may again assume that the simpler cases already hold, namely the following

$$\begin{aligned}(x' + y) + z &= x' + (y + z)\\ (x + y') + z &= x + (y' + z)\\ (x + y) + z' &= x + (y + z')\end{aligned}$$

where we may introduce even more primes into the equations if we want to, but it is not necessary for this proof. We also need to know what $(x + y)'$ means; namely an element of $x + y$, note that

by definition this is of the form $x' + y$ or $x + y'$. With all of this we again get a proof that consists of just stitching together the right definitions and induction hypotheses [2, 7].

$$\begin{aligned}
(x+y)+z &= \bigcup_{\substack{(x+y)'\in x+y\\ z'\in z}} \{(x+y)'+z, (x+y)+z'\} \\
&= \bigcup_{\substack{x'\in x\\ y'\in y\\ z'\in z}} \{(x'+y)+z, (x+y')+z, (x+y)+z'\} \\
&= \bigcup_{\substack{x'\in x\\ y'\in y\\ z'\in z}} \{x'+(y+z), x+(y'+z).x+(y+z')\} \\
&= \bigcup_{\substack{x'\in x\\ (y+z)'\in y+z}} \{x'+(y+z), x+(y+z)'\} \\
&= x+(y+z).
\end{aligned}$$

From line one to line two, we see that we changed $(x+y)'$ into two parts, namely $x'+y$ and $x+y'$ as was justified in the above argument. Now all parts of line two are in the form where we can apply the induction hypothesis. From there on we reverse our steps to get $x+(y+z)$.

We would also like to write this whole equation stitching using our shorthand notation, just to get a feel of how the notation works. The proof then becomes

$$\begin{aligned}
(x+y)+z &= \{(x+y)'+z,\ (x+y)+z'\} \\
&= \{(x'+y)+z,\ (x+y')+z,\ (x+y)+z'\} \\
&= \{x'+(y+z),\ x+(y'+z),\ x+(y+z')\} \\
&= \{x'+(y+z),\ x+(y+z)'\} \\
&= x+(y+z).
\end{aligned}$$

□

We can now freely drop the parentheses around multiple additions of games, since $x+y+z$ will yield the same answer, regardless of which way the additions are done.

Lemma 10. *With 0 representing the empty game $\varnothing$, we have for all games x that $0+x = x+0 = x$.*

Proof. There are no elements in the empty set, so a union over the empty set is empty itself. This means that $x+0 = \bigcup_{x'\in x}\{x'+0\}$. However, by the induction hypothesis, we know that $x'+0 = x'$. This means $\bigcup_{x'\in x}\{x'+0\} = \bigcup_{x'\in x}\{x'\} = x$. We get $0+x = x$ by commutativity. □

I hope the reader is now used to our shorthand notation as to not be confused by it anymore, as from now on we will no longer state the unions explicitly.

Notation. Whenever we write $\{x'\}$, it is implied that we take the union over all $x' \in x$. That is to say $\bigcup_{x'\in x}\{x'\} \overset{\text{notation}}{=} \{x'\}$. We will sometimes still write the union explicitly for the sake of clarity.

We have proven a lot about the structure of addition over games. All these results lead to the following theorem.

Theorem 11. *Games have a commutative monoid structure.*

Proof. This theorem is a direct result of Lemmas 7, 8, 9 and 10. □

A bit of caution is needed here. Note that what is written here is that games have a monoid structure and not that they are a monoid. This is because the collection of all games form a proper class and not a set, as noted by Conway in [2] page 50, but we can still make it a set by setting some restrictions which we will not discuss for all impartial games, but we will discuss it for nimbers.

3.3 Multiplication of games

We will give be brief description of multiplication of games. Since multiplication on impartial games does not give a rich structure unless modifications are made, which we will discuss after the introduction of the nimbers. For now we will only state the definition and some lemmas.

Definition 12. *Multiplication of games* is defined as follows

$$x \times y := \{(x' \times y) + (x \times y') + (x' \times y')\}.$$

Lemma 13. *For any games x and y, we have that multiplication is commutative, so $x \times y = y \times x$.*

Lemma 14. *With 0 denoting the empty set, we have $0 \times x = x \times 0 = 0$.*

Lemma 15. *For any game x, we have $\{0\} \times x = x \times \{0\} = x$.*

3.4 The Nimbers

Now that we know that games are a set containing the states a player can go to, we can define the nimbers (yes, this is a pun on numbers and the game Nim). Consider a single pile Nim game. The moves a player can make are to reduce the pile of n object to a pile of $n' < n$ objects. This leads us to define nimbers inductively as follows.

Definition 16. A *nimber* $*n$ is the set of previous (and smaller) nimbers. So

$$*n = \{*0, *1, *2, \ldots, *(n-1)\} = *(n-1) \cup \{*(n-1)\}$$

where $*0 = \{\}$. The set of all finite nimbers is denoted by $\mathbf{On}_2$.

Most of the time the notation 0 and $*$ are used for the nimbers $*0$ and $*1$. We will use this convention too.

Some keen observers might have noticed that the definition of the nimbers exactly corresponds with the definition of Von Neumann ordinals. Indeed, in the last chapter we will explore transfinite nimbers, but we will proceed with the finite for now. Also note that there is a (well-founded) order on the nimbers induced by the identification between a nimber and the corresponding natural number or the ordinal numbers. We call the set of all finite nimbers $\mathbf{On}_2$, although $\mathbf{On}_2$ is sometimes used as the class of all ordinal numbers with Nim addition and multiplication like in On Numbers and Games.

Since all nimbers except 0 contain 0 (the empty set), it is possible for the first player to win immediately, and since 0 leaves no moves left for the first player, it is a win for the second player. This is an important property to keep in mind; player two can win if we have the 0 Nim pile. Things might change if we start adding different sizes of Nim piles to each other.

We have our nimbers and we have addition of games. What would happen if we were to add two different nimbers?

Example 17. We will see which sets we get when we compute $0 + 0$, $0 + *$ and $* + *$. First we note that $0 = \{\}$, hence we do not have any elements to consider when we want to compute $0' + 0$. This leads to the conclusion that $0 + 0 = \{\} = 0$. For $0 + *$ we get only one element in our set, we get the set $\{0 + 0\} = \{0\}$. This is exactly the definition of $*$, hence $0 + * = *$. Now, the last sum is a bit curious. We get the set $\{0 + *, * + 0\} = \{*, *\} = \{*\}$.

As seen in the example, the game $* + *$ is not a nimber itself. But maybe it is similar to a nimber. How would a game like this behave? One way to visualize this is by starting with an empty board. Then the first player can create a Nim pile of size 1, to which player two immediately responds by taking that one coin away. Player one cannot make a move now and loses. This game looks a lot like the zero game, in the sense that it is the second player who wins. We will show that this can indeed be made into an equality by introducing the mex function.

3.5 Minimum excluded value

Before we define the mex function, we have to know a bit more about the nimbers. Note that the nimbers have a correspondence to the natural numbers by identifying $*n$ with n. This also induces an order on the nimbers (hence the use of smaller in the definition of nimbers). It is this order that we use to define the minimum excluded value.

Definition 18. The *minimum excluded value* mex is defined as follows. Let S be a proper subset of $\mathbf{On}_2$; the nimbers. Then

$$\begin{aligned} \text{mex} : \mathcal{P}(\mathbf{On}_2) \setminus \mathbf{On}_2 &\to \mathbf{On}_2 \\ S &\mapsto \text{the smallest nimber value not in } S. \end{aligned}$$

This map is sometimes also called Grundy's function or Sprague-Grundy's function and is then denoted with a capital G like in [8] and [4], but I chose to use mex because when the game consists of just nimbers, then the mex function will literally be the *minimum excluded value*. This notation was also used by Conway in [1] and by Mark Jeeninga in his master thesis [5]. We restricted the inputs of the mex function to proper subsets of the nimbers, but more generally, the mex can be defined for all proper subset of impartial games and all proper subsets of all ordinal nimbers, but more on that in Chapter 7.

We will show some properties of the mex function. Firstly, it maps a nimber to the nimber itself. This is because $*n$ is the set of all nimbers that came before it. Thus, the smallest element that is not in the set $\{0, *, *2, \dots, *(n-1)\}$ is the element $*n$. The function also maps sets which contain nimbers that are not nimbers themselves to the smallest element not in that set. This is what was meant with really being the minimum excluded value.

3.6 Addition of Nimbers

Which the mex, we can make a better addition operation on the nimbers. Instead of taking just the game sum $+$, we take the mex of the Nim sum. This ensures that the result will again be another nimber.

Definition 19. The *Nim sum* $\oplus$ is recursively defined as follows

$$*n \oplus *m := \text{mex}\Big(\bigcup_{\substack{*n' \in *n \\ *m' \in *m}} \{*n' \oplus *m, *n \oplus *m'\} \Big) \overset{\text{notation}}{=} \text{mex}(\{*n' \oplus *m, *n \oplus *m'\}).$$

There are many interesting properties that this addition operation has. First of all, it is commutative. It is also associative, we can use backwards induction to prove both claims. Most proofs are taken directly from Conway's On Numbers and Games [2] or Knuth's Surreal Numbers [7]. It will be stated when the proofs deviate from theirs.

Throughout this chapter we will simplify notation by not writing the outer most parenthesis for the mex function.

Lemma 20. *Nim addition is commutative.*

Proof. This proof is quite similar to the proof of commutativity of game addition. We again have the induction hypotheses that $x' \oplus y$ and $x \oplus y'$ commute. From that we can make the following equalities

$$x \oplus y = \text{mex}\{x' \oplus y, x \oplus y'\} = \text{mex}\{x \oplus y', x' \oplus y\} = y \oplus x.$$

□

Lemma 21. *For all nimbers $*n$ and $*m$, the Nim sum $*n \oplus *m$ is less than or equal to the nimber $*(n+m)$.*

Proof. By the induction hypothesis, we can assume that $*n' \oplus *m \leq *(n'+m)$ and $*n \oplus *m' \leq *(n+m')$, where n' and m' are used to denote the natural numbers that correspond to $*n'$ and $*m'$. We know that $*n \oplus *m = \text{mex}\{*n' \oplus *m, *n \oplus *m'\}$. Since both values inside the set take on values less than or equal to $*(n'+m)$ or $*(n+m')$, so we can be sure that $*(n+m)$ is not an element of the set, and hence it is an upper bound for our minimum excluded value, which implies $*n \oplus *m \leq *(n+m)$. □

Corollary 22. $0 \oplus 0 = 0$.

Proof. This is a direct consequence of the previous lemma, as $0 \oplus 0$ has to be less than or equal to 0, and there is only one nimber that can satisfy that, namely 0 itself. □

For upcoming proofs we need to deal with the object of the form $\text{mex}\{(x \oplus y)'\}$, which can become quite annoying when not dealt with.

Lemma 23. *For any nimber x, y z and w, we have*

$$\mathit{mex}\{w, z \oplus \big(\mathit{mex}\{x' \oplus y, x \oplus y'\}\big)'\} = \mathit{mex}\{w, z \oplus (x' \oplus y), z \oplus (x \oplus y')\}.$$

Proof. We know that $\{x' \oplus y, x \oplus y\}$ contains at least all values less than $x \oplus y$. This means that

$$\text{mex}\{w, z \oplus (x' \oplus y), z \oplus (x \oplus y')\} \geq \text{mex}\{w, z \oplus \big(\text{mex}\{x' \oplus y, x \oplus y'\}\big)'\}$$

At the same time we know that all values for $x' \oplus y$ and $x \oplus y$ bigger than $x \oplus y$ yield different values when added to z, so that means that we must have equality.

□

Lemma 24. *Nim addition is associative.*

Proof. By our induction hypothesis, we may assume $x' \oplus (y \oplus z) = (x' \oplus y) \oplus z$, and two other but similar equations. We also need to know what $(y \oplus z)'$ means. This is a bit different to the regular case of $(y+z)'$ since there now is a mex function we need to consider, but we can use the previous lemma to deal with it.

$$\begin{aligned}
x \oplus (y \oplus z) &= \text{mex}\{x' \oplus (y \oplus z), x \oplus (y \oplus z)'\} \\
&= \text{mex}\{x' \oplus (y \oplus z), x \oplus \big(\text{mex}\{y' \oplus z, y \oplus z'\}\big)'\} \\
&= \text{mex}\{x' \oplus (y \oplus z), x \oplus (y' \oplus z), x \oplus (y \oplus z')\} \\
&= \text{mex}\{(x' \oplus y) \oplus z, (x \oplus y') \oplus z, (x \oplus y) \oplus z'\} \\
&= \text{mex}\{(x \oplus y) \oplus z', (x \oplus y)' \oplus z\} \\
&= (x \oplus y) \oplus z.
\end{aligned}$$

□

3.7 Nimbers are/as a group

With addition of nimbers, we might ask if there is more structure behind it, and there is. The set of all (finite) nimbers form an Abelian group. We know that nimber addition is a closed operation and that it is commutative and associative. We only need to show that there is some unit element and that every nimber has a additive inverse.

Lemma 25. *The nimber zero is the unit element of the group. That is to say*

$$x \oplus 0 = 0 \oplus x = x, \quad \forall x \in \boldsymbol{On_2}.$$

Proof. Our induction hypothesis is that $x' \oplus 0 = x'$. We also need to use the fact that $x \oplus 0'$ is an empty set, since there are no elements to consider in $0'$. This leads to

$$x \oplus 0 = \text{mex}\{x' \oplus 0, x \oplus 0'\} = \text{mex}\{x'\} = x.$$

By commutativity we also get $0 \oplus x = x$. □

Something remarkable happens for the inverse of a nimber. When we play a game of Nim which consists of two equal stacks of coins, then the second player has a really easy strategy to win, namely to always mirror what the first player does. This indicates that the Nim sum $x \oplus x$ is 0, and it is. But before we can prove that we need the following lemma;

Lemma 26. *$x \oplus y = x \oplus z$ if and only if $y = z$.*

Proof. From right to left is easy to see, but from left to right requires a proof.

Suppose $y < z$ (or $y \in z$), then $x \oplus y$ is an element of $x \oplus z$. This contradicts that $x \oplus y = x \oplus z$, so $y \geq z$. Via an analogous argument we get $z \geq y$. This can only happen when $y = z$. □

Lemma 27. *For all nimbers we have*

$$x \oplus x = 0.$$

Proof. This is again a proof by backwards induction. This proof differs from the one in On Numbers and Games and Surreal Numbers [2] [7]:

$$x \oplus x = \text{mex}(\{x' \oplus x, x \oplus x'\}) = \text{mex}\{x' \oplus x\}.$$

By Lemma 26 which says that $x \oplus y = x \oplus z$ if and only if $y = z$, and our induction hypothesis, which says $x' \oplus x' = 0$, we have that 0 cannot be an element of $\{x' \oplus x\}$, since otherwise we have some specific $y \in x$ for which we have $y \oplus x = 0 = y \oplus y$ which implies $x = y$ which is a contradiction as y is an element of x.

Therefore 0 is the minimum excluded value of the set $\{x' \oplus x\}$, so $x \oplus x = 0$. □

With this proof completed, we've shown that the set of all finite nimbers under Nim addition is a group. Since we already had commutativity of addition, we know $(\mathbf{On_2}, \oplus, 0)$ is an Abelian group.

Binary exclusive OR

There is a really easy way to compute the Nim sum of two nimbers, because it turns out that this form of summation is exactly the same as using exclusive or on binary numbers. We set out to prove this in this section.

Lemma 28. *If $*n = \{0, *, *2, \ldots, *(n-1)\}$ is not a group and not empty, then there are some $*k, *m \in \{0, *, \ldots * (n-1)\}$ such that $*k \oplus *m = *n$.*

Proof. We know that 0 is in the set, which is the identity element. The inverse of any element is also in the set, because the inverse is the element itself. We also know that Nim addition is associative, so the only way the set could not be a group is by not being closed. This means that there exist some $*k, *m$ in the set such that $*k \oplus *m \geq *n$. If we have equality, then we are done, so suppose $*k \oplus *m = *l > *n$. By the definition of addition, $*l = \text{mex}\{*k' \oplus *m, *k \oplus *m'\}$. This means that there was an element of the form $*k' \oplus *m$ or $*k \oplus *m'$ equal to $*n$. Both $*k'$ and $*m'$ are part of the set, so we showed existence. □

Lemma 29. *Let $*n = \{0, *, \ldots, *(n-1)\}$ be a group. Then for all $m, k < n$, the sum $*(n \cdot m) \oplus *k = *(n \cdot m + k)$.*

Proof. This proof is from [2]. For our induction hypothesis we may assume that $*(n \cdot m') \oplus *k = *(n \cdot m' + k)$ and that $*(n \cdot m) \oplus *k' = *(n \cdot m + k')$ where a prime before a integer means that it can take any integer value less than it.

$*(n \cdot m) \oplus *k = \text{mex}\{*(n \cdot m)' \oplus *k, *(n \cdot m) \oplus *k'\}$. The elements that are excluded are $*(nm) \oplus *k'$ and $*(nm)' \oplus *k$.

We can use our induction hypothesis on the elements in the first form to see that we excluded $*(nm + k')$.

The second can be written as $*(n \cdot m' + n') \oplus *k$. Since $*n$ is a group, we have that $*n' \oplus *k = *i$ ranges over all values in the group. We can now apply our induction hypothesis twice to get

$$*(nm' + n') \oplus *k = *(nm') \oplus *n' \oplus *k = *(nm') \oplus *i = *(nm' + i),$$

where we remember that $*i$ ranges over all values in the group.

We see that we excluded exactly all values less than $*(nm+k)$, hence the minimum excluded value is $*(nm + k)$ itself.

□

This lemma implies that if a group has 2^i many element, that it behaves like binary exclusive or.

Theorem 30. *The Nimbers less than 2^i are groups*

Proof. Suppose we already know that $*2^i$ is a group. By Lemma 28, we need to find the first nimber after $*2^i$ that is closed under addition. We know by Lemma 29 that we can always make the next nimber using the previous nimbers by $*(2^i + k)$ for $k \in \{0, \ldots 2^i - 1\}$. We can do this until we hit $*(2^i \cdot 2)$. This cannot be created using previous nimbers. Any two nimbers smaller than $*2^i$ add to something still smaller than $*2^i$. If we have one that is smaller and one that is bigger or equal to $*2^i$ will still be smaller than 2^{i+1} by Lemma 29. Any two nimbers that are both bigger than or equal to $*2^i$ must be smaller than 2^i since $*(2^i + k_1) \oplus *(2^i + k_2) = *2^i \oplus *2^i \oplus *k_1 \oplus *k_2 = *k_1 \oplus *k_2$.

Therefore $*2^{i+1}$ is the next nimber that is closed under addition and thus a group. □

Theorem 31. *Nim addition is binary exclusive or. That is to say that every nimber can be written as a sum of powers of 2 using nim addition*

Proof. This theorem is a consequence of Lemma 29. We know that each number has a unique binary representation. So for any nimber $*k = *(2^{k_0} + 2^{k_1} + \ldots + 2^{k_m})$ where $k_0 > k_1 > \ldots k_m$. By normal induction with Lemma 29, we get that this is equal to $*2^{k_0} \oplus *2^{k_1} \oplus \ldots \oplus 2^{k_m}$. □

This theorem also has Lemma 21 as a corollary.

3.8 Multiplication

Multiplication of nimbers is inspired by the idea of trying to avoid zero divisors.

To avoid zero divisors, we will look at what a product needs to satisfy. If we have any two nonzero nimbers x and y then of course their product also must be nonzero. This also means that $(x - x')(y - y') \neq 0$. This relation can be written differently as $xy \neq x'y' + x'y - x'y'$. So any multiplication operation has to satisfy this relation. We look at all the values we cannot obtain and pick the smallest possible value that is left, and since in our case, addition is the same as subtraction, we get the following definition.

Definition 32. Let x and y be nimbers, then the *product* $x \circ y$ is defined as

$$x \circ y := \operatorname{mex}(\{(x' \circ y) \oplus (x \circ y') \oplus (x' \circ y')\}).$$

This sometimes gets written in more insightful notation, using the fact that $+$ is the same as $-$ which is the same as $\oplus$, to $\operatorname{mex}(\{x'y + xy' - x'y'\})$, to emphasise that it came from the above inequality. We will sometimes also drop the $\circ$ and instead mark multiplication by juxtaposition.

There are many properties that this multiplication has, and we will later show that this turns the nimbers less than $*2^{2^i}$ into a field. To do this, we first need to show that it is a commutative ring without zero divisors. So let us start with commutativity. All

Lemma 33. *Multiplication is commutative.*

Proof. This is once again a one line proof taken from [2]

$$\begin{aligned} x \circ y &= \operatorname{mex}\{x'y \oplus xy' \oplus x'y'\} \\ &= \operatorname{mex}\{yx' \oplus y'x \oplus y'x'\} \\ &= \operatorname{mex}\{y'x \oplus yx' \oplus y'x'\} \\ &= y \circ x. \end{aligned}$$

□

Lemma 34. *The nimber* 0 *behaves like the zero element under multiplication. That is to say,* $0 \circ x = 0$.

Proof. Remember that in order to use multiplication, we need elements of both the first and the second term. But 0 is the empty set, so when we try to write $0' \circ x$, we see that we do not have any elements. Therefore, any multiplication with 0 results in the mex of the empty set, which equals 0. Written out we get

$$0 \circ x = \operatorname{mex}(\emptyset) = 0.$$

□

Lemma 35. *The nimber* $*$ *is a multiplicative identity.*

Proof. We can use the fact that $0 \circ x = 0$ to get

$$x \circ * = \operatorname{mex}\{* \circ x' \oplus 0x \oplus 0x'\} = \operatorname{mex}\{x'\} = x.$$

□

Lemma 36. *Multiplication is associative.*

Proof. Proof by Conway.

For this proof we need to remember our induction hypothesis, namely that if we introduce a prime in any element, that we can use associativity, i.e. $x' \circ (y \circ z) = (x' \circ y) \circ z$. Also we need to remember once again what an element of $y \circ z$ entails

$$\begin{aligned}
(x \circ y) \circ z &= \operatorname{mex}\{(x \circ y)' \circ z \oplus (x \circ y) \circ z' \oplus (x \circ y)' \circ z'\} \\
&= \operatorname{mex}\{\left(\operatorname{mex}\{x'y \oplus xy' \oplus x'y'\}\right)' \circ z \oplus (x \circ y) \circ z' \oplus \left(\operatorname{mex}\{x'y \oplus xy' \oplus x'y'\}\right)' \circ z\} \\
&= \operatorname{mex}\{(x'y)z \oplus (xy')z \oplus (xy)z' \oplus (x'y')z \oplus (x'y)z' \oplus (xy)'z' \oplus (x'y')z'\} \quad \star \\
&= \operatorname{mex}\{x'(yz) \oplus x(y'z) \oplus x(yz') \oplus x'(y'z) \oplus x'(yz') \oplus x(y'z') \oplus x'(y'z')\} \\
&= \text{“} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{”} \\
&= x \circ (y \circ z).
\end{aligned}$$

□

The equality at $\star$ needs some justification. The main argument is that if we have any set of nimbers α with $\operatorname{mex}\{\alpha\} = x$ and any set β with $\operatorname{mex}\{\beta\} = y$ that we can still talk about $\alpha \circ \beta$ and similarly for $\alpha \oplus \beta$, and that with this we have $\operatorname{mex}\{\alpha'\beta \oplus \alpha\beta' \oplus \alpha'\beta'\} = xy$. One can make this more rigorous when using theorem 39 and 40.

Lemma 37. *Multiplication is distributive over addition.*

Proof. Proof by Conway.

$$\begin{aligned}
(x \oplus y) \circ z &= \operatorname{mex}\{(x \oplus y)' \circ z \oplus (x \oplus y) \circ z' \oplus (x \oplus y)' \circ z'\} \\
&= \operatorname{mex}\{(x' \oplus y)z \oplus (x \oplus y)z' \oplus (x' \oplus y)z', (x \oplus y')z \oplus (x \oplus y)z' \oplus (x \oplus y')z'\} \\
&= \operatorname{mex}\{x'z \oplus xz' \oplus x'z' \oplus yz, xz \oplus y'z \oplus yz' \oplus y'z'\} \\
&= \operatorname{mex}\{\left(\operatorname{mex}\{x'z \oplus xz' \oplus x'z'\}\right)' \oplus yz, xz \oplus \left(\operatorname{mex}\{y'z \oplus yz' \oplus y'z'\}\right)'\} \\
&= \operatorname{mex}\{(xz)' \oplus yz, xz \oplus (yz)'\} \\
&= xz + yz.
\end{aligned}$$

□

We again have used some variation of Lemma 23 and a similar argument that we had to use for $\star$.

Theorem 38. *There exist a multiplicative inverse and a inductive formula to find it.*

This is a really annoying theorem which we shall not prove. All it goes to show is that the nimbers form a field, something we will show in the next chapter without the use of a formula for the inverse.

3.9 mex for impartial games

We have addition of games and addition of nimbers, and they look similar apart from one using the mex function. Is there a relation between the two?

Theorem 39. *For two games x and y, we have $mex(x + y) = mex(x) \oplus mex(y)$.*

This allows us to speed up our calculations of addition of games. Similarly we have the following theorem.

Theorem 40. *For two games x and y, we have $mex(x \times y) = mex(x) \circ mex(y)$.*

4 The tower of fields

The main idea of this chapter is to show that there exists a tower of fields, namely $\mathbb{F}_2 \subset \mathbb{F}_4 \subset \mathbb{F}_{16} \subset \ldots$ that can be made by using adjoining a zero of an irreducible Artin-Schreier polynomial, which is a polynomial of the form $x^2 - x - \alpha$ (for some suitable α). Afterwards we show that there are isomorphisms from these fields to our nimber fields, and that $\mathbf{On}_2 = \bigcup_{i=0}^{\infty} \mathbb{F}_{2^{2^i}}$. We end by showing that there are no different fields in our nimbers.

As we have that $0 \oplus 0 = * \oplus * = 0$ and $0 \oplus * = * \oplus 0 = *$ together with $0 \circ * = * \circ 0 = 0 \circ 0 = 0$ and $* \circ * = *$, we can see that this is isomorphic to the field $\mathbb{F}_2 = \mathbb{F}_{2^{2^0}}$. If we look at the next few nimbers up to $*3$, we can see that we have the field $\mathbb{F}_4 = \mathbb{F}_{2^{2^1}}$. The next field that we can stumble upon is the one where we include all nimbers up to $*15$ which gets us to $\mathbb{F}_{16} = \mathbb{F}_{2^{2^2}}$. We will show that, starting from $\mathbb{F}_2$, we can construct all other fields that arise in the nimbers which are isomorphic to the fields of order a Fermat-2-power, which is the name we give to numbers of the form 2^{2^m}.

Notation. Throughout this chapter, we use the letters q and n, for which we have $n \in \mathbb{Z}_{\geq 0}$ and $q = 2^n$. We also want some shorthand notation for Fermat-2-powers, so we will use $r = 2^q = 2^{2^n}$.

4.1 The Frobenius map

In this section we will build towards finding a Artin-Schreier polynomial for $\mathbb{F}_q$, which is used to extend $\mathbb{F}_q$ to $\mathbb{F}_{q^2}$. To do this we need to start from the basics. We know that the Frobenius map on a field of characteristic 2 is the map

$$\begin{aligned} \mathcal{F} : \mathbb{F}_q &\to \mathbb{F}_q \\ x &\mapsto x^2 \end{aligned}$$

which is a field homomorphism. We can check that for any two elements a and b in $\mathbb{F}_q$, that $\mathcal{F}$ preserves multiplication, addition and that it sends the unit element to itself. The proof of multiplication is straightforward. Using commutativity we get $\mathcal{F}(a \cdot b) = (a \cdot b)^2 = a^2 \cdot b^2 = \mathcal{F}(a) \cdot \mathcal{F}(b)$. For multiplication we need to use that we work in a field of characteristic 2, then using distributivity and commutativity we get $\mathcal{F}(a + b) = (a + b)^2 = a^2 + 2a \cdot b + b \cdot a + b^2 = a^2 + a \cdot b + a \cdot b + b^2 = a^2 + b^2 = \mathcal{F}(a) \cdot \mathcal{F}(b)$.

The Frobenius map will be the the starting point form which we will construct our other maps. It is therefore important that we go over a few of its properties; the most important one being $\mathbb{F}_2$-linearity. When we have a field $\mathbb{F}_q$, we can see it as a vector space over over $\mathbb{F}_2$, the Frobenius map becomes a linear transformation over that. This will then be used to construct the different map $\wp$, called the Artin-Schreier map.

We need to show later on that the Artin-Schreier map is $\mathbb{F}_2$-linear, so we start with showing the Frobenius map is $\mathbb{F}_2$-linear.

Lemma 41. *The map $\mathcal{F} : \mathbb{F}_q \to \mathbb{F}_q$ is $\mathbb{F}_2$-linear.*

Proof. In order to prove that a map is linear over a field, we need to show that it preserves the vector addition and scalar multiplication. The addition was already proved by $\mathcal{F}$ being a homomorphism. The preservation of scalar multiplication can be proven using the fact that multiplication itself is preserved, and then restricting it to the elements 0 and 1.

$\mathcal{F}(1 \cdot n) = \mathcal{F}(1) \cdot \mathcal{F}(a) = 1 \cdot \mathcal{F}(a)$

$\mathcal{F}(0 \cdot n) = \mathcal{F}(0) \cdot \mathcal{F}(a) = 0 \cdot \mathcal{F}(a)$. □

4.2 The Artin-Schreier map

One of me more important maps on the fields $\mathbb{F}_q$, is the Artin-Schreier map, which is the map

$$\begin{aligned}\wp : \mathbb{F}_q &\to \mathbb{F}_q \\ x &\mapsto x^2 - x.\end{aligned}$$

The Artin-Schreier map is important as it allows us to find the right values for α for our $\mathcal{A}$-polynomials.

We would like to know what happens to our elements when we apply the Artin-Schreier map to it. Therefore we look at what the kernel of our map is.

Theorem 42. *The kernel of $\wp$ over $\mathbb{F}_q$ is $\{0, 1\}$.*

Proof. We know that $0^2 + 0 = 1^2 + 1 = 0$. These are two different solutions, and we cannot have more than two different zero's since the function is a degree two polynomial. Therefore the kernel is exactly $\{0, 1\}$. □

We also need to show that the Artin-Schreier map is $\mathbb{F}_2$-linear.

Theorem 43. *The map $\wp$ is linear over $\mathbb{F}_2$.*

Proof. The map $\wp$ can be expressed as $\mathcal{F} - id$. We already know that $\mathcal{F}$ was a $\mathbb{F}_2$-linear map. The identity is also linear, so this means that $\wp$ is a difference of two linear maps, which is still linear. □

Theorem 44. *The image of $\wp$ has $q/2$ many elements.*

Proof. When we look at $\wp$ as a linear map over $\mathbb{F}_q$, we can apply the rank-nullity theorem, which says that $\dim(\{0, 1\}) + \dim(\mathrm{Im}(\wp)) = n$. We know that $\dim(\{0, 1\}) = 1$ so the dimension of the image is $n - 1$, which implies that it has $\frac{1}{2}\#\mathbb{F}_q$ many elements. □

4.3 Irreducible Artin-Schreier polynomials

This is the section where we finally meet our $\mathcal{A}$-polynomials.

Definition 45. An irreducible Artin-Schreier polynomial or $\mathcal{A}$-polynomial for a field $\mathbb{F}_q$ is a polynomial of the form

$$x^2 - x - \alpha$$

for $\alpha \in \mathbb{F}_q$ such that it is irreducible in $\mathbb{F}_q[x]$. Note: normally any polynomial of the form

For what values of α do we get an $\mathcal{A}$-polynomial and can we systematically find such an α? It turns out that there are a lot of αs that satisfy the definition and that we can find such an α quite easily.

Theorem 46. *For $\alpha \in \mathbb{F}_q \setminus \wp(\mathbb{F}_q)$, the polynomial $x^2 - x - \alpha$ is irreducible in $\mathbb{F}_q$.*

Proof. Since we have a second degree polynomial, it suffices to check that we do not have any roots; since the only way a second degree polynomial could be reducible, is by it being the product of two linear factors. So suppose we do have a root of this polynomial β, then β satisfies

$$\beta^2 - \beta - \alpha = 0.$$

So this tells us that $\wp(\beta) = \alpha$. However, α was supposed to be an element that is was not hit by our map $\wp$, so this is a contradiction. Therefore our assumption that we have a root is false, and we conclude that $x^2 - x - \alpha$ is a irreducible polynomial. □

We can use these polynomials go get field extensions from $\mathbb{F}_q$ to $\mathbb{F}_{q^2}$.

This allows us to write a recursive procedure to get new field extensions. We can start with $\mathbb{F}_2$, and join the root of the polynomial $x^2 - x - 1$ to it. It is an irreducible polynomial since it was not in the image of $\wp$. We can call one of these roots x_1, and we get the field $\mathbb{F}_2(x_1)$. This new field contains the elements x_1 and $x_1 + 1$. We now need to find an element that is not in the image of $\wp$ applied to this field. We can systematically find one.

Definition 47. We define $\mathbb{F}_2(x_1, \ldots, x_{n+1})$ as the extension form $\mathbb{F}_2(x_1, \ldots, x_n)$ to which we add a zero of the polynomial $X^2 - X - \alpha_n$ for some suitable $\alpha_n \in \mathbb{F}_2(x_1, \ldots, x_n)$ that makes the polynomial irreducible.

This means that in order to talk about the field $\mathbb{F}_2(x_1, \ldots, x_{n+1})$, we must already have defined $\mathbb{F}_2(x_1, \ldots, x_n)$ and we should also state what α_n is.

This next theorem is based on Theorem 4.2 in M. Jeninga's master thesis [5] in which he shows a way to make many different towers, one of them being the tower we're currently working on. The proof uses some results from algebra that we first will take for granted, afterwards we will prove the results that we used.

Theorem 48. $\alpha_n = \begin{cases} \prod_{j=1}^{n} x_j & n \geq 1 \\ 1 & n = 0. \end{cases}$ *is an element in* $\mathbb{F}_2(x_1, x_2, \ldots, x_n)$ *that is not in the image of* $\wp$.

Proof. This proof is due to M. Jeninga [5].
The first claim we make it that α_n not being in the image of $\wp(\mathbb{F}_2(x_1, \ldots, x_n))$ is equivalent to applying the Artin-Schreier map $2^n - 1 = q - 1$ times and getting 1 as a result. We will write superscripts above our map to indicate repeated applications. So written out we get

$$\wp^{[q-1]}(\alpha_n) = 1 \Leftrightarrow \alpha_m \notin \wp(\mathbb{F}_2(x_1, \ldots, x_n)).$$

Now, We start a prove by normal strong induction. For the base case we know that 1 is not in $\wp(\mathbb{F}_2)$. Now for the induction hypothesis. We assume that for all $k \leq n$, $\prod_{i=1}^{k} x_i$ is not in the image of $\wp$ over $\mathbb{F}_2(x_1, x_2, \ldots, x_k)$. Now we want to show that when we extend the field using $x^2 - x - \prod_{i=1}^{n} x_i$, that the resulting field $\mathbb{F}_2(x_1, x_2, \ldots, x_n, x_{n+1})$, has $\prod_{i=1}^{n+1} x_i$ not in the image of $\wp$.

Consider $\wp^{[2^{n+1}-1]}(\alpha_{n+1})$, this is the same as $\wp^{[q-1]}(\wp^{[q]}(\alpha_{n+1}))$. Another claim that we will back up later is that $\wp^{[2^n]}$ is $\mathbb{F}_2(x_1, \ldots, x_n)$-linear. This allows us to write

$$\wp^{[q-1]}(\wp^{[q]}(\alpha_{n+1})) = \wp^{[q-1]}(\alpha_n \cdot \wp^{[q]}(x_{n+1})).$$

We know, by the definition of our previous field extension, that $x_{n+1}^2 - x_{n+1} - \alpha_n = 0$. If we multiply both sides by x_{n+1}^{-1} and rearrange, we get $x_{n+1} = 1 + x_{n+1}^{-1} \cdot \alpha_n$. This means

$$\wp^{[q]}(x_{n+1}) = \wp^{[q]}(1 + x_{n+1}^{-1} \cdot \alpha_n) = \wp^{[q]}(1) + \wp^{[q]}(x_{n+1}^{-1} \cdot \alpha_n) = \alpha_n \cdot \wp^{[q]}(x_{n+1}^{-1}),$$

where we again used the fact that $\wp^{[q]}$ is linear over $\mathbb{F}_2(x_1, \cdots x_n)$ to pull out the α_n.
Now we need to know what $\wp^{[q]}(x_{n+1}^{-1})$ is. Again consider $x_{n+1}^2 - x_{n+1} - \alpha_n = 0$. If we divide by $x_{n+1}^2 \cdot \alpha_n$, we get

$$\alpha_n^{-1} - \alpha_n^{-1} \cdot x_{n+1}^{-1} - (x_{n+1}^{-1})^2 = 0.$$

This means that x_{n+1}^{-1} is a root of $X^2 - \alpha_n^{-1} X + \alpha_n^{-1}$. Since x_{n+1}^{-1} was not in the field $\mathbb{F}_2(x_1, \ldots, x_n)$, it cannot be the solution of a linear polynomial. Therefore this has to be the minimum polynomial. This then implies that $\wp^{[q]}(x_{n+1}^{-1}) = \alpha_n^{-1}$.

If we now substitute all equations we see that

$$\wp^{[2^{n+1}-1]}(\alpha_{n+1}) = \wp^{[q-1]}(\alpha_n) = 1.$$

□

There were some claims that need backing up. We first try to justify why $\wp^{[q]}$ has the properties that we said it has.

Lemma 49. *$\wp^{[2^n]}$ is the map that sends and element x to $x^{2^{2^n}} - x$. In other words, $\wp^{[q]} = \mathcal{F}^q - id$.*

Proof. We know that $\wp$ is the map $x \mapsto x^2 - x$. using proof by induction, we can first assume $\wp^{[2^i]}$ is the map $x \mapsto x^{2^{2^i}} - x$. Then the map $\wp^{[2^{i+1}]}$ is the map $\wp^{[2^i]}$ applied twice. We thus get

$$\left(x^{2^{2^i}} - 1\right)^{2^{2^i}} - \left(x^{2^{2^i}} - 1\right).$$

Then, due to the binomial theorem, the fact that we are working on a field of characteristic 2, we get that this is equal to $x^{2^{2^{i+1}}} - x$ □

Lemma 50. *The map $\mathcal{F}^i : \mathbb{F}_q \to \mathbb{F}_q$, $x \mapsto x^{2^i}$ is an automorphism of $\mathbb{F}_q$, for $i \in \mathbb{Z}_{\geq 0}$.*

Proof. We know that $\mathcal{F}$ is a field homomorphism from $\mathbb{F}_q$ to itself. Composing a homomorphism with another one yield again a homomorphism. This means that all the mappings from x to x^{2^i} are field homomorphisms. It turns out that all of these homomorphisms are in fact automorphisms. Consider the kernel of $\mathcal{F}$, it is an ideal of $\mathbb{F}_q$. A field has only two ideals, $\{0\}$ and $\mathbb{F}_q$ itself. We know that $\mathcal{F}(1) = 1$, thus the kernel is $\{0\}$. This means that $\mathcal{F}$ is injective, and since $\mathbb{F}_q$ is a finite set, the mapping also has to be surjective. This means that we have an automorphism. □

Lemma 51. *$\mathcal{F}^{[n]}$ is $\mathbb{F}_q$-linear.*

Proof. We first note that any nonzero element in $\mathbb{F}_q$ has an order dividing $q - 1$. This is because of the multiplicative group being cyclic. This means that $\mathcal{F}^{[n]}$ is the identity map, which is $\mathbb{F}_q$-linear. □

With this last lemma, we have justified one of the claims we used in the proof of Theorem 48. Now we need to prove the other claim.

Lemma 52. $\wp^{[2^n-1]}(\alpha_n) = 1 \Leftrightarrow \alpha_n \notin \wp(\mathbb{F}_2(x_1, \ldots, x_n))$.

Proof. Let β be a root of $X^2 - X - \alpha_n$. Then $\wp^{[2^n-1]}(\alpha_n) = \wp^{[2^n]}(\beta) = \beta^{2^{2^n}} - \beta$. It turns out that $\beta^{2^{2^n}} - \beta = 0$ if and only if $\beta \in \mathbb{F}_2(x_1, \ldots x_n)$ which is equivalent to $\alpha_n \in \wp(\mathbb{F}_2(x_1, \ldots, x_n))$. This implies that $\alpha_n \notin \wp(\mathbb{F}_2(x_1, \ldots x_n)) \Leftrightarrow \wp^{[2^n-1]}(\alpha_n) \neq 0$, but since $\wp^{[2^n-1]} : \mathbb{F}_{2^{2^n}} \to \mathbb{F}_2$, we must have that it is 1. □

We once again shoved the proof of this claim, onto a different one which we will immediately prove.

Lemma 53. *All elements $a \in \mathbb{F}_q$ satisfy $a^q - a = 0$, and elements $b \in \mathbb{F}_{q^2}/\mathbb{F}_q$ have $b^q - b \neq 0$.*

Proof. We know that the the units of $\mathbb{F}_q$ form a cyclic group under multiplication and that the order of that group is $q - 1$. This means that $a^q = a$. For 0, we can immediately see that $0^q = 0$.

If β is an element of $\mathbb{F}_{q^2}/\mathbb{F}_q$, the the order of β in the multiplicative group of $\mathbb{F}_{q^2}$ divides q^2, but not q, so it must be q^2. This means that $\beta^q \neq \beta$, hence $\beta^q - \beta \neq 0$. □

With this, we have finally covered all there is to Theorem 48.

We now need to show that $\mathbf{On}_2$ is isomorphic to this tower of fields, to to this we show that every subfield of $\mathbf{On}_2$ is isomorphic to a field in the tower

Theorem 54. *$\mathbb{F}_2(x_1, \ldots, x_m)$ is isomorphic to the field of all nimbers less than $*2^{2^m}$, by mapping x_i to $*2^{2^i-1}$ for all $i \leq m$.*

Proof. This proof is due to Conway in [2].
This is also a proof by backwards induction. Suppose we already knew $*2^{2^{m-1}}$ (which are all nimbers less than it) is a field. By the uniqueness of finite fields, it is isomorphic to $\mathbb{F}_2(x_1, \dots x_{m-1})$. We suppose that every nimber $*2^{2^i}$ really corresponds to x_{i+1} for all $i \leq m-1$.

To continue we actually need some of the theorems Conway proved in On Numbers and Games, namely his "Simple extension theorems", for which we already have proven the first two; Lemma 28 and 29. The one that we need, Theorem 45, p58, states that if a nimber is a field but not algebraically closed (which our finite fields are), then it is a root of the "lexicographically earliest" polynomial with no root in the field. This theorem will not be proven in this thesis. It turns out that the Artin-Schreier polynomial with the corresponing value of α_{m-1}, is the lexicographically earliest polynomial not with a root in $*2^{2^{m-1}}$.

Next, Conway's 46th theorem (also part of his simple extension theorems) tells us that the addition multiplication done in this bigger field like we expect it. It tells us that we can evaluate any nimber in a field like $*2^{2^n}$ as follows.

$$(*2^{2^n} \circ *a_1) \oplus *a_0 = *(2^{2^n} \cdot a_1 + a_0)$$

for $*a_1, *a_0 \in 2^{2^n}$. This means that we can really view our x_i-s as corresponding to the right Fermat-2-power.

□

Corollary 55. *For any n, we have* $\left(*2^{2^n}\right)^2 = *(\frac{3}{2} \cdot 2^{2^n})$.

Proof. We know that $*2^{2^n}$ corresponds to x_{n+1}, which satisfies $X^2 = X + \alpha_n$. This means that

$$\left(*2^{2^n}\right)^2 = * \left(2^{2^n} \cdot \prod_{j=1}^{n-1} 2^{2^j} \right) = * \left(\frac{3}{2} \cdot 2^{2^n} \right)$$

. □

Lemma 56. *$x^q - x$ has no multiple zeros.*

Proof. We know that the derivative of $x^q - x$ is equal to $qx^{q-1} - 1$. But we are working in a field of characteristic 2, so the derivative becomes just -1. This means that there are no multiple zeroes. □

Lemma 57. *$\mathbb{F}_q$ is the splitting field of the polynomial $x^q - x$ over $\mathbb{F}_2$.*

Proof. This proof is from [9] Since the polynomial has no multiple zeros, the splitting field must have the elements $\{a_0, a_1, \dots, a_q\}$ all distinct and zeroes of $X^q - X$. But these zeroes also satisfy that $(a_i + a_j)^q - (a_i + a_j) = 0$ by the Frobenius homomorphism, and $(a_i a_j)^q - (a_i a_j) = 0$ and $(a_i^{-1})^q - (a_i^{-1})$ for all i and $j \in \{0, \dots, q\}$. This means that the set $\{a_0, \dots, a_q\}$ is actually a subfield of the splitting field. But the splitting field $\mathbb{F}_2(a_0, \dots a_q)$ is a subset of $\{a_0, a_1, \dots, a_q\}$, hence this must be the splitting field.

□

Corollary 58. *The tower of fields $\mathbb{F}_2 \subset \mathbb{F}_4 \subset \mathbb{F}_{16} \subset \dots$ is unique up to isomorphism.*

This corollary is due to the fact that splitting fields are unique up to isomorphism.

The last corollary also implies that our Nimbers $\mathbf{On}_2$ which is equal to this tower, is also unique.

4.4 Galois groups

It turns out that every extension $\mathbb{F}_2(x_1, \dots x_{n+1})/\mathbb{F}_2(x_1, \dots x_n)$ is Galois. We can show this by checking if the polynomial that defined the extension splits in the bigger field and if it has distinct zeroes in it.

Lemma 59. *The polynomial $X^2 - X - \alpha_n \in \mathbb{F}_2(x_1, \dots x_n)[X]$ splits in $\mathbb{F}_2(x_1, \dots, x_{n+1})[X]$ and has distinct roots x_{n+1} and $x_{n+1} + 1$.*

Proof. We can check that $(X + x_{n+1})(X + x_{n+1} + 1)$ gives us our polynomial back. We need to use that $x_{n+1}^2 = x_{n+1} + \alpha_n$. □

We can now see that the extension is indeed Galois. But how about the extension from $\mathbb{F}_2$ to $\mathbb{F}_2(x_1, \dots x_n)$?

Since every field extension we took was due to adding a zero of an irreducible polynomial of order 2, we have that the extension $\mathbb{F}_2(x_1, \dots x_{n+1})/\mathbb{F}_2(x_1, \dots x_{n+1})$ is Galois. This is because the polynomial now splits in our new field.

Theorem 60. *The extension $\mathbb{F}_2(x_1, \dots, x_n)/\mathbb{F}_2$ is Galois.*

Proof. Since we already new that splitting field s are unique, we know that $\mathbb{F}_2(x_1, \dots, x_m)$ is isomorphic to the splitting field of $X^{2^{2^n}} - X$ over $\mathbb{F}_2$. We showed $X^{2^{2^n}} - X$ was separable hence the extension is Galois. □

What do these Galois groups look like? The automorphism group $\mathrm{Aut}(\mathbb{F}_q)$ consist of different powers of the Frobenius map, until it gets back to being the identity map. So it is $\{\mathrm{id}, \mathcal{F}, \mathcal{F}^{[2]}, \dots, \mathcal{F}^{[q-1]}\}$. This is naturally isomorphic to $\mathbb{Z}/q\mathbb{Z}$.

5 Programming nimber addition and multiplication

We discovered a lot about the structure that the nimbers have, but we have not done any explicit calculations. If we wanted to know what $(*2022 \circ *1729) \oplus *31415$ is by hand, that would be really tedious. In this chapter, we will implement two different algorithms for addition and two different algorithms for multiplication of nimbers. We will also compute some polynomial equations using these algorithms.

This section will use Magma because is a powerful programming language which can be used for free via the Magma site. It also has no limits on the size of integers you can use.

5.1 The original definitions coded

Before we can implement the code for the original definition, we need to code the max function. For finite nimbers it is quite easy to compute. Given a set, we check if 0 is in the set, and if it isn't we check for the next number, and then the next, until we find a number that is not in the set. This number is the minimum excluded value of the set.

```
[mex]
mex:= function(S);
i := 0;
while i in S do
    i := i + 1;
end while;
return i;
end function;
```

This is a reasonably fast program. It should be $\mathcal{O}(n)$ in time at worst, with n being the size of the set.

Now we can implement addition. Since addition is defined recursively, our program would be too.

```
[Addition]
Add := function(n,m);
Output := {}; // This will be the final set that we put through the max function
x := {0..n-1}; // construct the nimber
y := {0..m-1};
for z in x do
    Output := Output join {$$(z,m)}; // inductively add x' + y to the set
end for;
for w in y do
    Output := Output join {$$(n,w)};
end for;
return mex(Output); // the mex of Output is the nimber we are after
end function;
```

This algorithm is spectacularly bad. It takes a lot of time to compute the sum of small numbers. Below I plotted a graph in Matlab that shows the relation between the calculation of `Add(10,n)` for $n \in \{0, \ldots, 7\}$ and the average time it takes for the online browser version of Magma to compute that value 5 different times, rounded to two decimals after the point. The small sample size is due to the fact that the online version of Magma stops computing whenever it has been running for two minutes. This happens for the calculation `Add(10,8)`, hence it cannot be plotted on the graph.

The values I found for $n = 0, \ldots, 7$ are as follows

n	0	1	2	3	4	5	6	7
time	0.01	0.03	0.12	0.46	1.73	6.22	20.78	67.05

Plotting this table gives When we divide the time value of n by the one of $n-1$, we see that

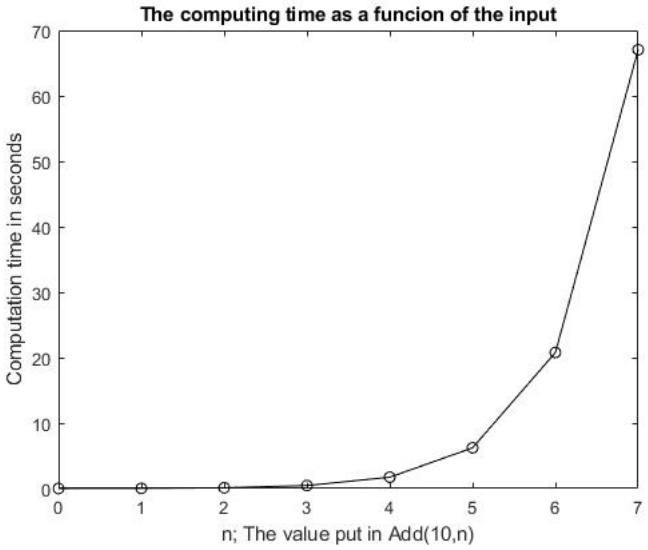

Figure 5: Computing time

we get exponential growth in time with a base between 3 and 4. Which I find surprising since the

recursive nature of the algorithm let's me to think that it should behave like a factorial. Undeniably however is that the main reason why this algorithm is so slow, is because of its recursive nature.

5.2 Faster algorithms

If we want to have a faster algorithm, then we need something that does away with the recursion, but we know of a different way to add two nimbers, namely to binary exclusive or them. We can thus make faster algorithms doing exactly this.

```
Add := function(n,m);
if m le n then // if m is smaller than n, swap them around
    a := m;
    m := n;
    n := a;
end if;
x := Intseq(n,2); // the binary repesitation of n (in reverse order)
y := Intseq(m,2);
i := #x;
j := #y;
for k in {i+1..j} do // filling in the missing zeroes
    x[k] := 0;        // so the sequences have the same lenght
end for;

for k in {1..j} do // XOR
    x[k] := Abs(x[k] - y[k]);
end for;
Output := 0;
for k in {1..j} do // making the final output number
    if x[k] eq 1 then
        Output := Output + 2^(k-1); // Magma starts indexing at 1
    end if;
end for;

return Output;
end function;
```

This is a really fast algorithm. It is not recursive anymore which allows it to compute additions of nimbers with several digits. For a demonstration of its speed, I chose some arbitrary numbers between 0 and 10^{25} and computed their Nim sums.

a	b	$a \oplus b$
4847381782734080799239049	6778423099897418741533812	1931929433905992411129853
7333835062302402685418147	6229268323450549420702386	3890763134890839421190161
3547451684189401312708021	7027894805684716428932748	8760087635495893003531065
6012538830143602990075256	1679164417444447002363435	6774968944027081281915731
6626544573140098746378272	5996165599367199544398670	1881964114636314525973358

Any one of these calculations took the supercomputer less than 0.020 seconds.

We also want an algorithm that can calculate the Nim multiplication fast. This however cannot be done without recursions. Still, the algorithm was made is pretty fast. It is also quite long so this algorithm will be put in the appendix.

6 Nimber codes

In the book "The book of Numbers" by John Conway and Richard Guy [3], there is a remarkable fact about a code, namely the integral lexicographical code of minimum distance 3 called S_3, is closed under pointwise Nim addition and multiplication. However, the book does not provide a proof, although a more general statement about codes of this kind was proven earlier by Conway in a 1987 paper [1]. In this section we will restate that this specific code forms a vector space and try to prove as much as possible that, if we assume that S_3 is a vector space, that we get our Nim addition and multiplication back. What is a code? Normally a code consist of code words which

are finite sequences of elements usually belonging to some finite alphabet. We define code words a bit differently.

Definition 61. A *code word* is a sequence of non-negative integers $(a_i)_{i=0}^{\infty}$ such that for only finitely many $a_i \in \mathbb{Z}_{\geq 0}$ we have $a_i \neq 0$. We write the sequence (a_i) in reverse order, so

$$(a_i)_{i=0}^{\infty} = (\cdots, a_3, a_2, a_1, a_0).$$

Definition 62. The *distance* between two code words $(a_i)_{i=0}^{\infty}$ and $(b_i)_{i=0}^{\infty}$ is defined by

$$d((a_i)_{i=0}^{\infty}, (b_i)_{i=0}^{\infty}) = \sum_{i=0}^{\infty} ||a_i - b_i||_H \tag{1}$$

where $||x||_H$ is the Hamming norm on $\mathbb{Z}$; the norm that returns 0 if $x = 0$ and 1 otherwise. This distance function is also known as the Hamming distance.

Definition 63. For two different code words (a) and (b), we say $(a) < (b)$ if and only if $\exists n$ such that $a_n < b_n$ and $\forall N > n$, $a_N = b_N$.

This definition is the reason why we write our code words in reverse order. By doing that we can easily read which of the words is the smaller one by interpreting them as normal numbers, although we need to be careful that a single element of a word can take any value, not just one digit values.

Definition 64. The *integral lexicographic code of minimal distance* 3 is the code with the properties that the zero code word is in it, and for every code word (w), the set $\{(v) \leq (w)\}$ has the properties that

$$\begin{aligned} &i)\, d(w, v) \geq 3 \quad \forall v \\ &ii)\, \forall s \text{ s.t. } d(s, v) \geq 3\, \forall v \implies w \leq s. \end{aligned}$$

Just like the nimbers have a mex function, the code words have one too. This mex is a bit more involved than the one on nimbers since it has to deal with several infinitely long lists of code words, but it works the same way.

Theorem 65. *For two code words (a) and (b), we have*

$$(a) \oplus (b) = mex\{(a)' \oplus (b), (a) \oplus (b)'\},$$

and

$$\mu(a) = mex\{\mu'(a) \oplus \mu(a)' \oplus \mu'(a)'\},$$

where $(a)'$ ranges over all the code words that came before (a) and where μ is a nimber. This gives the same result as doing pointwise Nim addition.

Proof. The proof can be found in integral lexicographic codes [1]. □

Theorem 66. *These code words are closed under pointwise nimber addition and multiplication. They even form a vector space over the nimbers.*

Proof. This again can be found in integral lexicographic codes [1] □

Example 67. We want to see what this sequence of code words looks like. We have no code words to begin with so the first one we get is $(\ldots, 0, 0, 0)$. The next word we get is $(\ldots, 0, 0, 0, 1, 1, 1)$, then $(\ldots, 0, 0, 0, 2, 2, 2)$. If we continue we get$(\ldots, 0, 0, 0, n, n, n)$.

Lemma 68. *The first code words are all of the form* $(\ldots, 0, 0, 0, n, n, n)$.

Proof. This proof is from Conway's paper on codes [1]. It is a proof by (normal) induction.

Suppose we know the first m code words are all of the form $(\ldots, 0, 0, 0, n, n, n)$ for $n < m$, we want to show that the next code word is $(\ldots, 0, 0, 0, m+1, m+1, m+1)$. We should try to find the smallest word, so we can assume $a_i = 0$ for $i \geq 3$. What if $a_2 \leq m$, then there is some code that has at most distance 2 from it. Thus ruling out $a_2 \leq m$. The smallest option we can choose is $a_2 = m + 1$. We similarly come to the conclusion that $a_1 = m + 1$ and $a_0 = m + 1$. Thus $(\ldots, 0, 0, 0, m+1, m+1, m+1)$ is the next code word in our sequence. □

Although we now have infinitely many code words, we do not stop. We can set $a_3 = 1$ to look at what the next smallest code word would be. Since $a_2 = 0$ doesn't cause the word to be within distance 2 of other words, it becomes the smallest possible choice. With that, we cannot choose $a_1 = 0$ anymore because otherwise the word would be too close to $(\ldots, 0, 0, 0)$, so we choose $a_1 = 1$. And with that we are forced to pick $a_0 = 2$. We can continue to find the next code words by checking it with all previous words or prove the patterns that exist with induction again and again, but that is too tedious.

Theorem 69. *If we assume the integral lexicographic code is a vector space over some field with* $\mathbb{N}$ *as its elements but with addition and multiplication yet undefined, then the addition on* $\mathbb{N}$ *must be* $\oplus$*. Furthermore, if we assume* 1 *is the multiplicative identity, then we get* $\circ$ *as our multiplication.*

procedure to find the nimbers. From [1].
We know that the zero code word is in the code. If we add the zero code word to itself, we should get a code word consisting the same element, for all positions. This can then only be the zero code again. Thus $0 \oplus 0 = 0$.

This immediately tells us that 0 has to the additive identity of our field since no element except 0 satisfies $x \oplus x = x$.

We can similarly use code words higher in the list to see what the value is of any sum we might have.

For multiplication it is a bit different. If we do not assume 1 is the multiplicative identity, then we can make any element the multiplicative identity by defining a new multiplication operation that uses the previous one. namely $a \circ_{\text{new}} b = a \circ_{\text{old}} b \circ_{\text{old}} c^{-1}$. This makes c the new multiplicative identity.

So if we suppose that 1 is the multiplicative identity, we can again preform multiplications on the code words and see what values we should get. In this way, we get our nimbers.

7 Transfinite Nimbers

We have mainly discussed finite nimbers in this thesis. As hinted throughout, this is not all there is. Nimbers can be defined for any ordinal number. The mex function also extends to the ordinals. For example

$$\text{mex}\{0, *, *2, *3, \ldots\} = *\omega,$$

where ω is the first finite ordinal. Nimber addition can easily be extended to transfinite ordinals, as ordinals also have unique binary representations for which the rule of binary exclusive or still holds. Nimber multiplication however does not extend in a easy way.

There are many field to be found in the nimbers due to Conway's simple extension theorems. Fields like ω (which we have worked with in this thesis) but also ω^ω and even ω^{ω^ω}. Of the many field that are found, we only know that ω^{ω^ω} is algebraically closed. Finding the next algebraically closed field is tremendously hard task that as far as I know has not been solved yet.

8 Appendix

```
Multiply := function(n,m);
x := Intseq(n,2);  // construct base 2 repesentation
y := Intseq(m,2);
i := #x;
j := #y;
b := [];
d := [];
Output := 0;

Index := 1;          // We want to know the exponents of the powers of 2 that make up n
for k in {1..i} do   // also in base 2 representation
    if x[k] eq 1 then
        b[Index] := Intseq(k-1,2);
        Index := Index + 1;
    end if;
end for;

Index := 1;        // similarly but for m
for k in {1..j} do
    if y[k] eq 1 then
        d[Index] := Intseq(k-1,2);
        Index := Index + 1;
    end if;
end for;     // E.g now we have list of the form [[1,1],[1,0,1]] representing 40

h := 0;                 //we will make all elements (sequences) the same length
for k in {1..#b} do  // to do that we first need the maximum lenght
    h := Max({h,#b[k]});
end for;
for k in {1..#d} do
    h := Max({h,#d[k]});
end for;

for k in {1..#b} do   //now we change everything to have the max lenght
    for f in {#b[k]+1..h} do
        b[k,f] := 0;
    end for;
end for;

for k in {1..#d} do      //similarly for d
    for f in {#d[k]+1..h} do
        d[k,f] := 0;
    end for;
end for;

// Now the actual multiplication starts
```

```
for k in {1..#b} do // I won't indent these two for for clarity
for l in {1..#d} do
h := #b[k];
ManagableTerm := []; // these will contain different powers of 2^2^i
NonManagableTerm := []; // these will be our 2^2^i terms squared
EndTerm := 1;

// as long as there is not a 1 in the same spot for both, we can add the values

for f in {1..h} do
    if b[k,f] eq 1 and d[l,f] eq 1 then  // now we have to do 3*2^(2^n -1)
        ManagableTerm[f] := 0;
        NonManagableTerm[f] := 1; // noting that this term is a square of 2^2^i
    else;                                  // we can just use the value we have
        ManagableTerm[f] := b[k,f] + d[l,f];
        NonManagableTerm[f] := 0;
    end if;
end for;

// Now we can normally multiply the values in the managable term
for f in {1..h} do
    if ManagableTerm[f] eq 1 then
        EndTerm := EndTerm * 2^(2^(f-1));
    elif NonManagableTerm[f] eq 1 then
        EndTerm := $$(EndTerm,3*2^(2^(f-1) -1) ); // The only...
    end if; // ...recursive part of this algoritm, since we cannot compute this...
end for; // efficiently

Output := Add(Output, EndTerm);
end for;          // the end of our multiplication and indents
end for;

return Output;
end function;
```

References

[1] J. H. Conway. "Integral lexicographic codes". In: *Discrete Math.* 83.2-3 (1990), pp. 219–235. ISSN: 0012-365X. DOI: 10.1016/0012-365X(90)90008-6. URL: https://doi.org/10.1016/0012-365X(90)90008-6.

[2] J. H. Conway. *On numbers and games*. Second. A K Peters, Ltd., Natick, MA, 2001, pp. xii+242. ISBN: 9781568811277.

[3] J. H. Conway and R. K. Guy. *The book of numbers*. Copernicus, New York, 1996, pp. x+310. ISBN: 9780387979939. DOI: 10.1007/978-1-4612-4072-3. URL: https://doi.org/10.1007/978-1-4612-4072-3.

[4] Thomas S Ferguson. "GAME THEORY".

[5] M. Jeeninga. *On a Tower of Fields related to* $\boldsymbol{On}_p$. 2015.

[6] A. H. Jorgensen. "Context and driving forces in the development of the early computer game Nimbi". In: *IEEE Ann. Hist. Comput.* 31.3 (2009), pp. 44–53. ISSN: 1058-6180. DOI: 10.1109/MAHC.2009.41. URL: https://doi.org/10.1109/MAHC.2009.41.

[7] D. E. Knuth. *Surreal numbers*. Addison-Wesley Publishing Co., Reading, Mass.-London-Amsterdam, 1974, pp. iii+119. ISBN: 9780201038125.

[8] H. W. Lenstra Jr. “Nim multiplication”. In: *Séminaire de Théorie des Nombres 1977–1978*. CNRS, Talence, 1978, Exp. No. 11, 23.

[9] J. Top. *Algebraic Structures*. 2017.