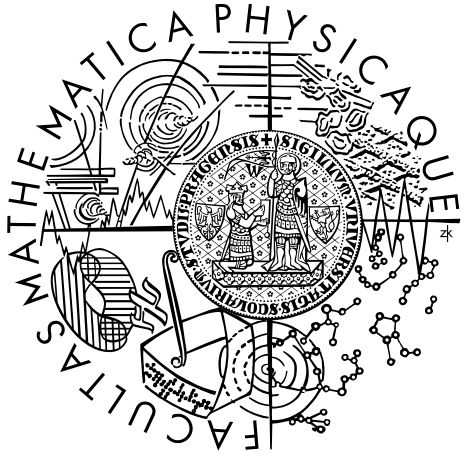
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Michal Hojsík

Proudová šifra RC4

Katedra Algebry

Vedoucí diplomové práce: *Doc. RNDr. Aleš Drápal, CSc.*

Studijní program: *Matematika,*
Matematické metody informační bezpečnosti

Prohlašuji, že jsem svou diplomovou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 20.4.2006 Michal Hojsík

Contents

Názov práce: Prúdová šifra RC4
Autor: Michal Hojsík
Katedra: Katedra Algebry
Vedúci diplomovej práce: Doc. RNDr. Aleš Drápal, CSc.
e-mail vedúceho: drapal@karlin.mff.cuni.cz

Abstrakt: Predložená práca sa zaoberá triedou zobecnených vnútorných stavov šifry RC4, tzv. zotrvalými stavmi. Prúdová šifra RC4 je najpoužívanejšou softvérovo založenou prúdovou šifrou a existencia takéhoto stavu by znamenala významnú slabinu tejto šifry. Pre zotrvalé stavy popisujeme tzv. tabuľkový model, pomocou ktorého dokazujeme periodičnosť týchto stavov. Ďalej študujeme vzťah medzi tabuľkovým modelom a ekvivalenciami na lineárne usporiadanej množine a dokazujeme regulárnosť matice zadanej ľubovoľnou z týchto ekvivalencií. Dosiahnuté výsledky uplatňujeme v teórii zotrvalých stavov RC4 a pre konkrétny prípad dokazujeme neexistenciu zotrvalého k-stavu pre k rovné $2, 3, 4$, ktorý by bol šifrou RC4 dosiahnuteľný. V práci taktiež popisujeme množstvo zotrvalých stavov, ktoré sú ale šifrou RC4 nedosiahnuteľné. V závere je načrtnuté využitie zotrvalých stavov v kryptoanalýze šifry RC4.

Kľúčové slová: prúdová šifra, RC4, vnútorný stav šifry

Title: The stream cipher RC4
Author: Michal Hojsík
Department: Department of Algebra
Supervisor: Doc. RNDr. Aleš Drápal, CSc.
Supervisor's e-mail address: drapal@karlin.mff.cuni.cz

Abstract: In the present work we study a class of generalised inner states of the cipher RC4, the so-called persistent states. The RC4 stream cipher is the most widely used software-based stream cipher and the existence of such a state would be a significant weakness of the cipher. We describe the Tabular model and using the model we prove the periodicity of these states. Then we study a new type of relationship between the tabular model and the equivalences on linearly ordered sets and we prove the regularity of the matrix determined by such an equivalence. Afterwards we apply the obtained result to the theory of persistent states and we prove that there exists no reachable persistent k-state for k equal to $2, 3, 4$ in the specific case. Moreover, we present some new unreachable persistent states. Finally, we indicate the cryptanalytical significance of the persistent states.

Keywords: RC4, stream cipher, the inner state

1 Introduction

The purpose of this work is to study a generalisation of profitable states, so called persistent states. In Section 2 we give a brief introduction into stream ciphers. In Section 3 we describe RC4 stream cipher and previous results about its security.

At the beginning of Section 4 we define several types of RC4 states. Afterwards we look closer at profitable and persistent states and show some basic properties of these states. In the end, we describe the Tabular model and table triples and using this model we prove that each persistent monotonous state has to be periodic. At the beginning of Section 5 we study a new type of relationship between equivalences on a linearly ordered sets and table triples, and prove several interesting theorems and propositions. Then we prove the nonexistence of monotonous persistent states of small orders. In the end, we focuse on cryptanalytical significance of persistent states.

2 Stream ciphers

Symmetric encryption algorithms (ciphers) can be generally divided into *block ciphers* and *stream ciphers*. Block ciphers (in the basic ECB mode) encrypt a group of characters (usually a block of bits of a plaintext message) using a fixed encryption transformation. Probably the best known block cipher DES (Data Encryption Standard, which is no more secure) has the block of length 64 bits, while its successor AES (Advanced encryption Standard) has the block of length 128 bits. On the other hand, stream ciphers encrypt individual characters using encryption transformation which varies in time (to make it more complicated, block ciphers can be used as stream ciphers when working in the CBC mode). Main reasons for using stream ciphers instead of block ciphers are the following:

1. stream ciphers are generally faster than block ciphers;
2. stream ciphers can process individual characters. This can be necessary when buffering is limited or when we need to decrypt characters as they are received (e.g. telecommunication applications);
3. stream ciphers have less complex hardware circuitry; and
4. stream ciphers have limited or no error propagation.

A general model of the stream cipher consists of an inner state of the cipher S, a next-state function f, a keystream producing function g and an output function h which combines keystream and plaintext to produce ciphertext. These functions have different inputs and design in different models of stream ciphers. The inner state of the cipher is some kind of memory of the cipher, it contains information that is necessary for encryption. The inner state usually changes in time and this change may be key-dependent. In some stream ciphers the key is used only for initial state generation and the next-state function is key independent. The keystream producing function generates a keystream which is later combined with a plaintext by the output function (this is where the adjective *stream* in "stream cipher" comes from).

The majority of currently used stream ciphers are the so called binary additive stream ciphers. That means they are operating on binary digits, and the output function h is the addition in GF(2), also known as the XOR function.

Since most of the stream ciphers used in practise tend to be proprietary and confidential, relatively few stream cipher designs are publicly accessible.

The most widespread design of stream ciphers is based on Linear Feedback Shift Registers (LFSR). The main reason for this is that they are well-suited for hardware implementation and that because of their structure they can be

readily analysed by using algebraic techniques. A5/1 and A5/2 are representatives of this class of stream ciphers. They are used in GSM networks for encryption of voice communication between the base station and the cell phone (both of these ciphers were kept in secret but were reverse engineered and afterwards broken).

Stream ciphers can be divided into synchronous and self-synchronising stream ciphers. We will look shortly on these types.

Synchronous stream ciphers: In this type of stream ciphers, the keystream is generated independently of plaintext and of ciphertext. This means that the sender and the receiver have to use the same key and operate at the same position (the same inner state) within that key to allow proper decryption. If this synchronisation is lost (e.g. by ciphertext character being deleted or inserted during the transmission) the decryption fails. On the other hand, if a ciphertext character is modified during the transmission (but not deleted), it does not affect the decryption of other ciphertext characters.

Let S, f, g, h denote the same as above and let k, c_i, m_i denote the key, the ciphertext character and the plaintext character where the subscript i means the position of the character within the stream (we can consider the ciphertext and the plaintext as streams of characters). S_i will denote the inner state at time i. The encryption process can be described as follows:

$$S_{i+1} = f(S_i, k),$$

$$z_i = g(S_i, k),$$

$$c_i = h(z_i, m_i).$$

The initial state S_0 is mostly derived out of the key. The encryption and decryption processes are illustrated in Figure 1. Remark that this is just a general model of a synchronous stream cipher and some synchronous ciphers do not match this model fully (i.e. in the RC4 stream cipher functions f and g are key-independent).

Self-synchronising (or asynchronous) stream ciphers : Here, the keystream is generated as a function of a key and a fixed part of previously generated ciphertext. Let k denote a key, g a keystream producing function, h an output function, z_i a keystream character, m_i a plaintext character, c_i a ciphertext character and let $S_0 = (c_{-t}, \ldots, c_{-1})$ be an initial state (the index i means as before the position of the character within a stream or a text). The encryption function of a self-synchronising stream ciphers can be described as follows (see Figure 2):

$$S_i = (c_{i-t}, c_{i-t+1}, \ldots, c_{i-1}),$$

$$z_i = g(S_i, k),$$

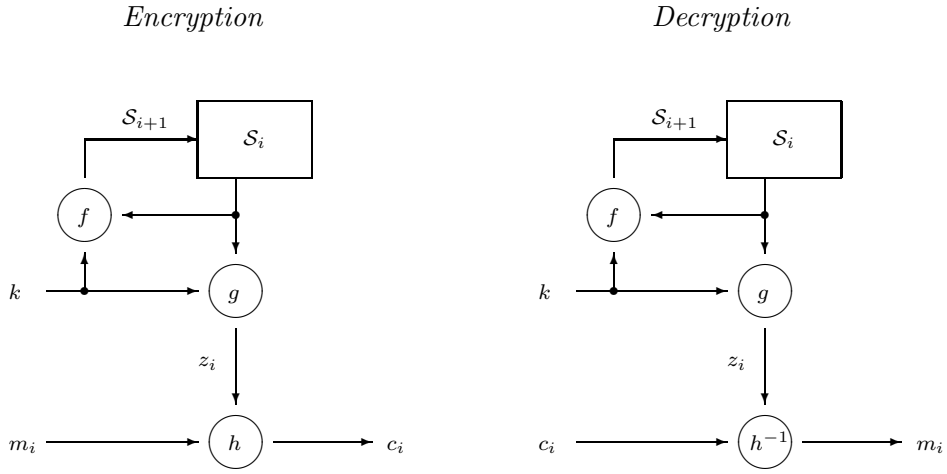


Figure 1: General model of a synchronous stream cipher

$$c_i = h(z_i, m_i).$$

Because the state of a self-synchronising stream cipher depends only on a fixed number of preceding ciphertext characters, the proper decryption can be still performed after some incorrect outputs, when a ciphertext character is modified or even deleted during transmission. This ciphertext dependence also makes these ciphers more resistant against attacks based on plaintext redundancy. It is because each plaintext character influences the entire following ciphertext and hence the statistical properties of the plaintext are distributed throughout the ciphertext.

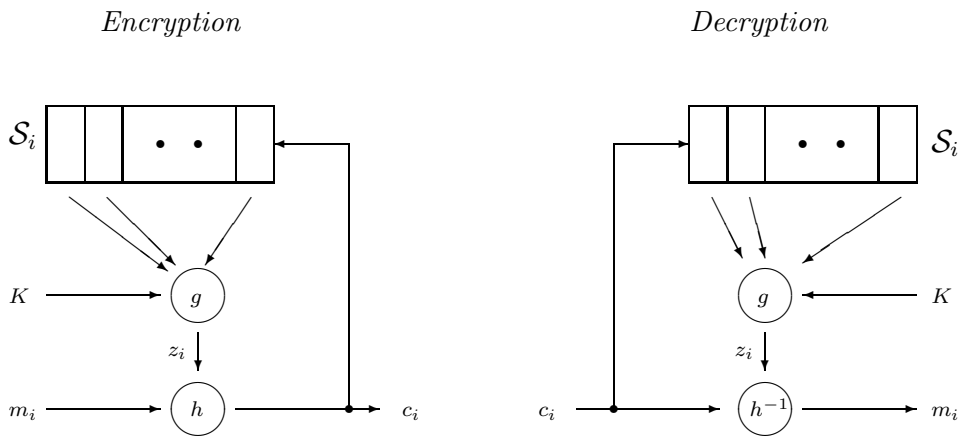


Figure 2: General model of an asynchronous stream cipher

3 The RC4 stream cipher

As mentioned above, the majority of published stream ciphers is based on various combinations of linear feedback shift registers which are easy to implement in hardware, but slow in software. So there was a need for a more software-friendly stream cipher. In 1987 Ron Rivest designed the RC4 stream cipher for RSA Data Security Inc. [1]. The cipher was implemented in products of RSA Data Security, but its specification was kept in secret. This has changed in 1994, when someone reversed-engineered the cipher and posted its source code in the C programing language to the Cipherpunks mailing list. The correctness of the description was confirmed by comparing its output to those produced by original RSA Data Security product. Later in their response ([RSA01]) to the IV attack ([SAR01]) based on this description, RSA Data Security Inc. provided a de facto approval for the correctness of this RC4 specification.

Nowadays, RC4 is the most widely used software based stream cipher. The most common use of RC4 is to protect the Internet traffic as a part of the SSL cipher suite, where it is commonly used as the default cipher for SSL/TLS connection. It was also implemented in products such as Microsoft Office, Lotus Notes, Oracle Secure SQL and many other applications.

3.1 Description of RC4

In the RC4 stream cipher simplicity of the design meets the security of the cipher, both on the high level.

RC4 is actually a class of algorithms parameterised by the size of its block. This parameter, n, is the word size for the algorithm. We will now describe $RC4_n$.

Let n be an integer and $N = 2^n$. The secret inner state of the cipher $RC4_n$ is a permutation $S \in \mathcal{S}_N$ of all N bit words and two indices $i, j \in \{0, \ldots, N-1\}$. The cipher is divided into two parts. In the first part, named Key Scheduling Algorithm (KSA), permutation S is derived out of an initial key (typical size is 40-256 bits). This is the only key-dependent part of the RC4. The second part called PRGA (Pseudo Random Generation Algorithm) uses this permutation to produce pseudo random bits (n in each round) and it contains the next-state function f and the keystream producing function g. The former continually shuffles the permutation S and the latter picks a value of the permutation S as an output (both these function are key-independent). Output bits are then bit-wise XOR-ed with the plaintext bits to produce the ciphertext, so RC4 is a binary additive synchronous stream cipher (e.g. output function h is binary addition).

[1] 'R' in the 'RC4' probably stays from Ron or Rivest and 'C' stays for 'code' or 'cipher'

Let K be an input key, S a permutation of the set $\{0, \ldots, N-1\}$ and let $K[i]$ denote the i-th n-bit block of K and $S[i]$ the value $S(i)$. The composition of permutation S with transposition (i, j) can be regarded as a swap operation if we represent the permutation S as a table with one row and N columns where the value $S(i)$ is in the i-th column.

An integer $l \in \mathbb{N}$ will denote the least integer equal to or exceeding the bit length of the key K divided by n. Thus $l = \lceil k/n \rceil$. Algorithm 1 and Algorithm 2 describe the KSA and the PRGA of $RC4_n$.

Algorithm 1 The Key Scheduling Algorithm of $RC4_n$

Input: secret key K
Output: permutation $S \in S_N$
$N \leftarrow 2^n$
$l \leftarrow$ number of n-bit blocks of K
for $i = 0$ to $N - 1$ **do**
    $S[i] = i$
end for
$j \leftarrow 0$
for $i = 0$ to $N - 1$ **do**
    $j \leftarrow (j + S[i] + K[i \bmod l]) \bmod N$
    Swap$(S[i], S[j])$
end for

Algorithm 2 The Pseudo Random Generation Algorithm of $RC4_n$

Input: permutation $S \in S_N$
Output: keystream
$N \leftarrow 2^n$
$i \leftarrow 0$
$j \leftarrow 0$
loop
    $i \leftarrow i + 1 \bmod N$
    $j \leftarrow (j + S[i]) \bmod N$
    Swap$(S[i], S[j])$
    Output $S[(S[i] + S[j] \bmod N)]$
end loop

Obviously the Key Scheduling Algorithm is a key-dependent variant of the keystream generation algorithm, which produces no output. We see that the input key length could be up to $n \cdot 2^n$ bits. Since the input key is used to generate only a permutation of 2^n values, the entropy provided by the key can be at most $\log_2(2^n!)$ bits. This is the effective key length.

In most applications, RC4 is used with the word length 8 bits (i.e. $n = 8$). That means that the inner state of the cipher consists of permutation S of 256 elements and of two indices $i, j \in \{0, \ldots, 255\}$. So we have 256! possibilities for S and 256^2 possibilities for i and j. Together it is approximately 2^{1700} possible inner states. Thus it is infeasible to guess even a small part of the inner state, or to use standard Time/Memory/Data tradeoff attacks [2], which are often applied against other stream ciphers. Furthermore, the inner state evolves in a nonlinear way, and thus it is difficult to use any partial information about the state far away in time. Consequently all known attacks against stream ciphers based on LFSR are inapplicable to RC4.

As mentioned above, the inner state of the RC4 with $n = 8$ can be represented with approximately 1700 bits. But on the other hand, the input key varies from 40 bits in the US export version of RC4 up to 256 bits. That is a huge difference. Where it comes from? The KSA part turns an input key into the permutation S, which is the predominant source of entropy, i.e. the main source of those 1700 bits. So logically, the KSA should be of great cryptanalytical interest. But behind the simplicity of the KSA algorithm and its clear mathematical description, the KSA is not easy to analyse and has received less attention than it should.

On the other hand, PRGA was given much more attention. Also our work has been focused mainly on PRGA.

3.2 Related Work

In 1995, Andrew Roos posted a paper [Ro95] to the sci.crypt newsgroup describing a class of weak keys for which the initial byte of the keystream is highly correlated with the first few key bytes. Weak keys are those satisfying the equality $K[0] + K[1] = 0 \mod N$. The weakness occur because there is a high probability (approximately $1/e$) that the KSA swaps a given entry of the s-box exactly one. This attack reduces the search effort by $2^{5.1}$, but if linearly related session keys are used (e.g. if there exists a linear relation between session keys), the reduction in effort can increase up to 2^{18}.

A sophisticated attack of RC4 is presented in [KMP98]. Authors developed cryptanalytic algorithms for a known plaintext attack where only a small segment of plaintext is assumed to be known. The attack is independent of the key scheduling algorithm and the key size. The idea behind the attack is to guess some part of the initial state of RC4, to detect incorrect guesses by looking for contradictions in the keystream, and finally to discover the rest of the initial state. The complexity of one of the attacks is estimated to be less that the time of searching through the square root of all possible initial

[2] Time/Memory/Data Tradeoff attack is some kind of the brute force attack were a speed up is reached by pre-computing of some values and storing them in memory.

states. However, this still poses no threat to RC4 in practical applications where $RC4_8$ is used. This attack is sometimes referred as Knudsen's attack and often serves as a final part of other attacks.

Mister and Tavares ([MT99]) developed several variants of a state backtracking attack. With their method, e.g., the inner state of $RC4_5$ can be obtained from a portion of the keystream using 2^{42} steps, while the nominal key space of $RC4_5$ is 2^{160}. But for the commonly used $RC4_8$ all of the proposed methods are infeasible. The authors of [MT99] also analysed properties of the state transition graph of RC4 states in the PRGA.

Grosul and Wallach analyse $RC4_8$ in related key attack [GW00]. They showed that for each 2048-bit input key (i.e. no repeating of the key is used in the KSA) there exists a family of related keys. These related keys produce the keystream, which is substantially similar in the initial hundred words before diverging. This attack has no practical application, since RC4 is commonly used with a 128 bits key and in the KSA this key is repeated to give the 2048 bit key which is needed to produce initial permutation.

Before mentioning publications about RC4 distinguishers, we will closely look at the notion of the distinguisher generally.

The best known unconditionally secret cipher is the Vernam cipher, also called the one-time-pad. In this cipher, a key (as long as a plaintext) is XOR-ed with the plaintext. The condition for the perfect secrecy is that the key has to be truly random and has to be as long as the plaintext. Otherwise we loose perfect secrecy.

Roughly speaking, additive synchronous stream ciphers are trying to simulate the Vernam cipher by producing a pseudo-random keystream (derived out of an input key), which is later XOR-ed with a plaintext. The more random the keystream is, the better security is achieved. However, since the keystream cannot be truly random (the amount of entropy is limited to the entropy of the input key since the cipher itself is deterministic), at some point in the keystream we should be able to distinguish between the stream cipher keystream generator (stream cipher) and a truly random keystream generator.

It follows from this logic that the more of keystream that is needed to distinguish it from a random output, the closer that keystream is to being random, and the better the cipher is. Therefore a common academic attack model for stream ciphers is the *distinguisher attack*. In this model, the goal is to come up with a distinguisher that can distinguish between the real cipher and a random output with as small amount of keystream as possible.

In [Go97] Golic proposed a linear model of $RC4_n$ using the linear sequential circuit approximation method. The model is successful because the permutation S evolves too slowly in the PRGA. This method requires $64^n/225$ keystream words and has correlation coefficient $15 \cdot 2^{-3n}$. The author estimates that this statistical defect allows an attacker to distinguish $RC4_8$ from

a random output after approximately 2^{40} successive output words.

Fluhrer and McGrew presented in [FM00] their observation of irregularities in the digraph distribution and they described a method for computing digraph probabilities in the PRGA. Using this method, they built a distinguisher for $RC4_8$ which requires $2^{30.6}$ output words. Furthermore they describe how to recover the parameters n and i if they are unknown. Authors were the first who mentioned the existence of special RC4 states (these states led them to the digraph distribution irregularities) and they named these states *fortuitous* states. Our work has been focused on *profitable* states which are a generalisation of *fortuitous* states.

The first half of the paper [FMS01] describes a set of weak keys in which a certain subset of key bits completely determines a subset of the output bits. Authors firstly define slightly modified version of KSA algorithm called KSA* , and identify a class of weak keys in which the knowledge of a small number of key bits suffices to determine many state and output bits. Afterwards they show that this happens also with unmodified KSA with a non-negligible probability. Authors use these weak keys to construct a new distinguisher for RC4, and to mount related key attacks with practical complexities.

Before looking on the second part of [FMS01], we will look on the notion of initialisation vector for a while.

One problem with binary additive stream ciphers is that the same key will always produce the same keystream. Of course, repeatly using the same key is just as bad as reusing a key in Vernam cipher, and it leads to the break of the cipher. Furthermore it is not necessary to know even the cipher specification in this case. It is enough to know that the keystream was used more than once. To avoid this problem, the concept of initialisation vector is useful.

An *initialisation vector* (IV) is a random value that is used to add some randomness to the output of the stream cipher. Since the IV has to be changed with every instance and it has to be random and unique, it makes the keystream different even if the same has been was used. There are many possibilities how to use IV to add randomness. In the most stream ciphers it is somehow combined with the input key and this combination is used as the key for the cipher. Because the (proper) receiver has to know the key used for the encryption for a proper decryption, it is necessary to transmit the IV to the receiver. IV is commonly transfered as it is (without any encryption), for example at the beginning of the ciphertext.

However, the use of IV can be a security weakness if not used properly. In some ciphers even a partial knowledge of the key together with some ciphertext can result in a substantial weakening. For such ciphers the simple concatenation of the IV to the input key, or XOR of the IV with the key can cause serious security flaw. RC4 suffers from this problem.

The second part of [FMS01] describes a weakness in the usage of RC4 in the

WEP (Wired Equivalent Privacy) protocol which has protected many wireless networks based on the IEEE 801.11b (Wi-Fi) standard. The security flaw rests in the way how the IV is used IV[3]. The authors described an IV weakness and showed how to use the IV weakness to attack systems that concatenate the IV before the key, concatenate the IV after the key, or XOR the IV and the key in order to produce the session-key used for encryption. Their attack easily discloses the secret key by analysing the first word of keystream generated from a small number of session-keys. This attack can be used also as ciphertext-only attack, while the first byte of the plaintext if often known or constant.

A practical usage of the attack presented in [FMS01] was described in [SAR01]. Paper describes the attack, its implementation and some optimalizations to make the attack more efficient. Authors were able to reconstruct an 128 bit secret key used in a network with a passive attack using their implementation and permission of the network administrator. The attack required roughly $5,000,000$ packets with the same key to determine the key. The conclusion is that the 802.11 WEP protocol is totally insecure. It is worth to mention that this insecurity is caused by a false usage of RC4 with IV, not by RC4 itself.

A lot of analysis of the probabilities of any given value being output by RC4 have been done. Most of these analyses have approached RC4 by looking on a given output. Mantin and Shamir, in [MS01], have had different approach. They have looked for polynomial-space, not polynomial-time distinguisher. (A polynomial-time distinguisher is trying to distinguish a keystream from a truly random output by taking a polynomially long keystream produced by one "run" of the cipher or of a random output. A polynomial-space distinguisher is given a black box which is either a true random generator or the cipher. The distinguisher can reset and rerun this black box with a random key a polynomial number of times and each of these keystreams has a fixed length.) Mantin and Shamir found out that the second word of RC4 keystream has a very strong bias. It takes the value 0 with twice the expected probability, e.g. $1/128$ instead $1/256$ for $n = 8$, while other values of the second output and all the values of other outputs have almost uniform distributions. Another contribution to the theory of special RC4 states can be also found in [MS01].

The aim of the paper [Mi02] is to find a conservative estimate for the number of bytes that should be discarded from the beginning of the $RC4_8$ keystream in order to reach as high level of security as possible. Based on his analysis, the author recommends discarding at least the first 512 bytes of an RC4 keystream.

Authors of [SP03] focus on a special set of RC4 states called *non-fortuitous predictive states*. Predictive states (for definition see Section 4) play a special

[3]The WEP uses 3 byte IV

role between all RC4 states. They increase the probability to determine a part of the inner state in a known plaintext attack and present a cryptanalytic weakness of RC4. Authors also formally proved the conjecture given by Mantin and Shamir in [Ma01] that, using only a known elements of the permutation S along with the two indices i and j at some round, the RC4 PRGA cannot produce more than a outputs in the next N rounds.

The second contribution of authors Souradyuti and Preneel can be found in [SP04a]. They have found a new statistical bias in the distribution of first two output bytes of $RC4_8$. The described distinguisher requires 2^{25} output bytes. This bias does not disappear even if the initial 256 bytes are dropped. Authors also present a new pseudo-random generator named RC4A, which works with two RC4 permutations.

The report [Wu05] describes a serious security flaw in Microsoft Word and Excel. Author found out that when a document encrypted by RC4 gets modified and saved again, the initialisation vector remains the same and thus the same RC4 keystream is applied to encrypt the different version of that document. As mentioned above, double use of the same keystream is crucial and a lot of information can be recovered easily from the document.

In [Ma05] author described a new statistical biases of the digraphs distribution of RC4/RC4A keystream, where digraphs tend to repeat with short gaps between them. These biases can be used to distinguish RC4 keystream of 2^{26} bytes and RC4A keystream of $2^{26.5}$ bytes. The second result presented in this paper is the discovery of so called *recyclable* states. The author uses these states and the state recovery attack from [KMP98] to mount the attack which recovers the inner state of the RC4 and requires approximately 2^{290} output bytes.

Also the trend of side-channel attacks has not skipped RC4. Efficient fault analysis attacks on RC4 are described in [BGN05]. One of these attacks is based on the usage profitable state discovered by Hal Finney ([Fi94]). We describe an extension of this attack in Section 5.4.

4 Special RC4 states

In this section we will focus on the PRGA part of the RC4 cipher. First of all, we will generally introduce the term k-state and mention predictive states. Then we will look at profitable states, which are the main object of our research. Afterwards we will introduce the notion of table triple and describe many properties of table triples. Finally, we will look at cryptanalytic applications of special class of profitable states, the persistent states.

Let $N \geq 1$ be an integer or $N = \infty$. Let $a, b \in \mathbb{Z}$. If $N \neq \infty$, then we will denote the integer $x \in \mathbb{Z}_N$ such that $x \equiv b \bmod N$ by $b \bmod N$. If $N = \infty$, then $a = b \bmod N$ means $a = b$. In the same manner, we define $\mathbb{Z}_\infty$ as $\mathbb{Z}$.

4.1 k-states, profitable states and predictive states

Let $i \in \mathbb{Z}_N$, $j \in \mathbb{Z}_N$ and let σ be a partial permutation of $\mathbb{Z}_N$ (a partial mapping of $\mathbb{Z}_N$ to $\mathbb{Z}_N$ that maps every two different elements from the domain into two different values). Assume that $\mathrm{Im}(\sigma)$ (the range of the mapping σ) is a finite set. This assumption is nontrivial only if $N = \infty$. The domain of the mapping σ will be denoted by $\mathrm{Dom}(\sigma)$.

Definition 1. *Let i, j, σ be as above and let $|Dom(\sigma)| = k > 0$. A* k-state *is any triple $T = (i, j, \sigma)$. We call T a* state*, when the value k is not emphasised.*

In the rest of this paper we will always consider only partial states (k-states with $k << N$), if not stated otherwise. Hence we will usually refer to a partial state as to a state.

It is clear that the *k-state* from Definition 1 is indeed a partially defined $RC4_n$ state (of PRGA) in the case $N = 2^n < \infty$.

The PRGA (Algorithm 2 in Section 3.1) describes a developement of the inner state of RC4. In the terms of Section 2 it describes the next-state function f and the keystream producing function g, which are both key independent. The following definition describes how far we can get when the PRGA is applied to a partial state and when we ignore the keystream producing function.

Definition 2. *Let $T_0 = (i_0, j_0, \sigma_0)$ be a state. Define a sequence $T_0, T_1, \ldots$ (which can be finite or infinite) in the following way: Let $T_k = (i_k, j_k, \sigma_k)$. If $i_k + 1 \notin Dom(\sigma_k)$, then T_k is the last term of the sequence. If $i_k + 1 \in Dom(\sigma_k)$, define $T_{k+1} = (i_{k+1}, j_{k+1}, \sigma_{k+1})$ in the following way:*

$$i_{k+1} = i_k + 1 \ mod\ N,$$

$$j_{k+1} = j_k + \sigma_k(i_{k+1}) \ mod\ N$$

$$\sigma_{k+1}(x) = \begin{cases} \sigma_k(i_{k+1}), & if\ x = j_{k+1} \\ \sigma_k(j_{k+1}), & if\ x = i_{k+1}\ and\ j_{k+1} \in Dom(\sigma_k) \\ not\ defined, & if\ x = i_{k+1}\ and\ j_{k+1} \notin Dom(\sigma_k) \\ \sigma_k(x), & elsewhere \end{cases}$$

If the resulting sequence is finite and equals $T_0, \ldots, T_h$, we say that the state T_0 has height h *or that the state T_0 is* h-profitable*. If the resulting sequence is infinite, we set height of the state T_0 to infinity and we say that T_0 is* persistent*.*

Let us observe what is happening in the previous definition. We do not insist upon $i_k \in \mathrm{Dom}(\sigma_k)$, but $i_k + 1 \in \mathrm{Dom}(\sigma_k)$ is the condition under which the sequence can be prolonged. Now, the image of $i_k + 1$ determines together with j_k the value j_{k+1}. The image of $i_k + 1$ becomes the new image of j_{k+1}. Again, $i_k + 1$, which was in $\mathrm{Dom}(\sigma_k)$, does not have to be in $\mathrm{Dom}(\sigma_{k+1})$. It is there if and only if $j_{k+1} \in \mathrm{Dom}(\sigma_k)$.

Remark 3. *Our definition of the* h-profitable k-state *is equal to the definition of the* weak h-profitable k-state *presented in* [Ma01] *for the case $N = 2^n < \infty$.*

For completeness, we will also mention *predictive* states, although our research has been focused on *profitable* and *persistent* states.

Definition 4. *Let $T_0 = (i_0, j_0, \sigma_0)$ be a k-state of height at least h, i.e. it induces the sequence $T_0, T_1, \ldots, T_h$, and let $A \subseteq \{1, \ldots, h\}$, $|A| = b$. If $(\sigma_l(i_l) + \sigma_l(j_l)\ mod\ N) \in Dom(\sigma_l)$ for all $l \in A$, then T_0 is said to be* b-predictive*.*

Assume we have a b-predictive k-state $T_0 = (i_0, j_0, \sigma_0)$ ($N = 2^n < \infty$). Then the value $z_l = \sigma_l(\sigma_l(i_l) + \sigma_l(j_l) \bmod N)$ is defined for all $l \in A$. According to Algorithm 2 z_l is the output value of RC4 in the round l. Therefore all full $RC4_n$ states (i, j, S) ($|\mathrm{Dom}(S)| = N$) compatible with T_0 (i.e. $i = i_0, j = j_0$ and σ_0 is a restriction of S) have the same output in all rounds determined by the set A, i.e. T_0 determines these output independently of the rest of the permutation S. That is why these states are called predictable states.

Mantin and Shamir stated in [Ma01] a conjecture that there exists no *b-predictive a-state* for $a < b << N$. This conjecture was later proved in [SP03].

Example 5. Figure 3 shows a 6-profitable 3-state for $N \geq 8$. This state is also 3-predictive with $A = \{4, 5, 6\}$.

We will now mention some cryptanalytic applications of predictive states as described in [Ma01].

The class of predictive states can be used to build distinguishers for RC4 keystream. More precisely, a b-predictive k-state implies the existence of a distinguisher for RC4 keystream which requires $O(N^{2k-b+3})$ output words.

i	j	$S[N-2]$	$S[N-1]$	$S[0]$	$S[1]$	$S[2]$	$S[3]$	$S[i]+S[j]$	Output
$N-3$	$N-1$	1	2	*	$N-1$	*	*	*	*
$N-2$	0	*	2	1	$N-1$	*	*	*	*
$N-1$	2	*	*	1	$N-1$	2	*	*	*
0	3	*	*	*	$N-1$	2	1	*	*
1	2	*	*	*	2	$N-1$	1	1	2
2	1	*	*	*	$N-1$	2	1	1	$N-1$
3	2	*	*	*	$N-1$	1	2	3	2

Figure 3: A 3-predictive 3-state (all additions are carried out *mod* N)

Another way how to use predictive states to attack RC4 is to use these states to determine a part of the inner permutation S. This can be done since all states compatible with a b-predictive k-state produce the same output. So an external event (an output sequence produced by an RC4 state) helps us to determine some information about the internal state of RC4. See [Ma01] for further details.

4.2 Closer look at profitable and persistent states

Notation: Suppose that T_l is a state. In the following text, we write Σ_l for $Im(\sigma_l)$. Sometimes we shall also write $k \in \{a, \ldots, h\}$ for the situation when h can be equal to ∞. This should be interpreted as $k \in \{i \geq a \mid a \in \mathbb{Z}\}$.

Lemma 6. *Let T_0 be a state of height h. Then $\Sigma_k = \Sigma_0$ holds for all $k \in \{0, \ldots, h\}$.*

Proof: If follows from Definition 2 that mappings σ_l and σ_{l+1} of two following states (these are T_l and T_{l+1}) have different values only for elements i_{l+1} and j_{l+1}. Because these values are only swapped, there is no difference between Σ_l and Σ_{l+1}. □

Hence we can write just Σ instead of Σ_0.

Lemma 7. *Let T_0 be a state of height h. Then $j_k \in Dom(\sigma_k)$ and $\sigma_k(j_k) = \sigma_{k-1}(i_k)$ holds for all $k \in \{1, \ldots, h\}$.*

Proof: Assume $k \in \{1, \ldots, h\}$. According to Definition 2, T_k is defined only if $i_k \in \mathrm{Dom}(\sigma_{k-1})$ and then $\sigma_k(j_k) = \sigma_{k-1}(i_k)$. Hence $j_k \in \mathrm{Dom}(\sigma_k)$. □

Although we start with the initial state T_0, we will sometimes also consider sequences starting with a state T_r, $r \in \mathbb{Z}$. Note that results obtained for T_0 can be easily transfered to such a situation.

Definition 8. *A state T_0 of height h is said to be* monotonous *if $\sigma_k(i_k)$ is not defined for any $k \in \{1, \ldots, h\}$, i.e. $i_k \notin Dom(\sigma_k)$ for all $k \in \{1, \ldots, h\}$. Clearly $i_k \notin Dom(\sigma_k)$ if and only if $j_k \notin Dom(\sigma_{k-1})$.*

Definition 9. *A state $T' = (i', j', \sigma')$ is said to be a* restriction *of a state $T = (i, j, \sigma)$, if $i' = i$, $j' = j$ and σ' is a restriction of σ.*

Soon we will prove that in the case $N = \infty$ we can obtain a monotonous persistent state from each persistent state by a restriction of one of its successors.

Definition 10. *Let T_0 be a persistent state. Define the value $\eta_k(a), k \geq 0$ for all $a \in \Sigma$ so that $\eta_k(a)+k$ is equal to the smallest $k' > k$ for which $\sigma_{k'}(j_{k'}) = a$. Set $\eta_k(a) = \infty$, if there is no such k'.*

We see that the value $\eta_k(a)$ is defined as a distance between the state T_k and the state $T_{k'}$ in the sequence of the states where $\sigma_{k'}(j_{k'}) = a$ and k' is the smallest such an integer.

Notation: If σ is a partial permutation of a set Ω, $a = \sigma(b) \in Im(\sigma)$, then $\sigma^{(-a)}$ will denote a restriction of the mapping σ to the set $\mathrm{Dom}(\sigma) \setminus \{b\}$.

Proposition 11. *Let $T_0 = (i_0, j_0, \sigma_0)$ be a persistent state. Assume that $\eta_k(a) = \infty$ holds for some $a \in \Sigma$ and $k \geq 0$. Then $(i_k, j_k, \sigma_k^{(-a)})$ is also a persistent state.*

Proof: Assume that k_0 is the smallest k for which $\eta_k(a) = \infty$. Without loss of generality we can assume that $k_0 = 0$. So for all $k \geq 1$, $\sigma_k(j_k) \neq a$ holds and thus $\sigma_k(i_{k+1}) \neq a$ holds for all $k \geq 0$. To prove the proposition, it is enough to prove that the state $(i_{k+1}, j_{k+1}, \sigma_{k+1}^{(-a)})$ is the successor of the state $(i_k, j_k, \sigma_k^{(-a)})$ for all $k \geq 0$.

We know that $\sigma_k(i_{k+1}) \neq a$. If $j_{k+1} \notin \mathrm{Dom}(\sigma_k)$ or $\sigma_k(j_{k+1}) \neq a$, then $\sigma_k^{-1}(a) = \sigma_{k+1}^{-1}(a)$, and $\sigma_{k+1}^{(-a)}$ is clearly the successor of $\sigma_k^{(-a)}$. Let $\sigma_k(j_{k+1}) = a$. Then $\sigma_{k+1}(i_{k+1}) = a$, and $\sigma_k^{(-a)}$ is not defined for j_{k+1}, and $\sigma_{k+1}^{(-a)}$ is not defined for i_{k+1}. Since this is not in contradiction with Definition 2, $\sigma_{k+1}^{(-a)}$ is the successor of $\sigma_k^{(-a)}$. □

Definition 12. *A persistent state T_0 is said to be* reduced, *if $\eta_k(a)$ is a finite positive integer for all $k \geq 0$ and for all $a \in \Sigma$.*

We have already seen that it is possible to derive a reduced persistent state T' from each persistent state T. In such a case the new range is a subset of the old one, i.e. $\Sigma' \subseteq \Sigma$.

Lemma 13. *Let $N = \infty$ and let $T_0 = (i_0, j_0, \sigma_0)$ be a persistent state. Assume that for some $k \geq 0$ there exists $j \leq i_k$ which is an element of $Dom(\sigma_k)$, and set $a = \sigma_k(j)$. Then $\eta_k(a) = \infty$.*

Proof: Denote by k_0 the smallest k, for which $\sigma_k^{-1}(a) \leq i_k$. Without loss of generality we can assume that $k_0 = 0$. For all $k \geq 0$ set

$$\Gamma_k = \{a \in \Sigma \mid \sigma_k^{-1}(a) > i_k\}.$$

First of all, we will show that $\Gamma_{k+1} \subseteq \Gamma_k$ for all $k \geq 0$. If $b \in \Gamma_{k+1}$ and $b \notin \{\sigma_{k+1}(j_{k+1}), \sigma_{k+1}(i_{k+1})\}$, then $\sigma_{k+1}^{-1}(b) > i_{k+1} > i_k$ and $b \in \Gamma_k$. If $b = \sigma_{k+1}(i_{k+1})$, then $\sigma_{k+1}^{-1}(b) = i_{k+1}$ and $b \notin \Gamma_{k+1}$. Finally, $\sigma_{k+1}(j_{k+1})$ is always an element of Γ_k since $\sigma_{k+1}(j_{k+1}) = \sigma_k(i_{k+1})$ and $i_{k+1} = i_k + 1 > i_k$.

In the beginning of this proof we have assumed that $\sigma_0^{-1}(a) \leq i_0$. Hence $a \notin \Gamma_0$. This implies that $a \notin \Gamma_k$ for all $k \geq 0$. Consequently $a \neq \sigma_{k+1}(j_{k+1})$ since $\sigma_{k+1}(j_{k+1}) = \sigma_k(i_{k+1})$ and $\sigma_k(i_{k+1}) \in \Gamma_k$. Finally, $\eta_k(a) = \infty$. □

Proposition 14. *Assume that $N = \infty$ and $T_0 = (i_0, j_0, \sigma_0)$ is a reduced persistent state. Then $\sigma_k^{-1}(a) > i_k$ holds for all $a \in \Sigma$ and for all $k \geq 0$. In particular, T_0 is monotonous.*

Proof: Define again Γ_k as $\Gamma_k = \{a \in \Sigma \mid \sigma_k^{-1}(a) > i_k\}$. Lemma 13 and Proposition 11 imply that $\Gamma_k = \Sigma$ holds for all $k \geq 0$. If $i_{k+1} \in \mathrm{Dom}(\sigma_{k+1})$, then for $a = \sigma_{k+1}(i_{k+1})$ we get $a \notin \Gamma_{k+1}$. Consequently T_0 is monotonous. □

Assume that T_0 is a reduced monotonous persistent state (with N finite or infinite). Lemma 7 implies that we can assume $j_0 \in \mathrm{Dom}(\sigma_0)$ without loss of generality. We can also assume that $i_0 \notin \mathrm{Dom}(\sigma_0)$, because $i_k \notin \mathrm{Dom}(\sigma_k)$ for all $k \geq 1$.

Lemma 15. *Let $T_0 = (i_0, j_0, \sigma_0)$ be a reduced monotonous persistent state ($N \in \mathbb{N}$ or $N = \infty$), $i_0 \notin Dom(\sigma_0)$ and $j_0 \in Dom(\sigma_0)$. Then $\eta_k(\sigma_k(j_k)) = j_k - i_k \bmod N$ and $0 < \eta_k(\sigma_k(j_k)) < N$ holds for all $k \geq 0$.*

Proof:

Set $a = \sigma_k^{-1}(j_k)$ and $d = j_k - i_k \bmod N$, $0 \leq d < N$. We have $j_k = i_k$ in the case $d = 0$. This implies $i_k \in \mathrm{Dom}(\sigma_k)$ which contradicts the monotonous property of T_0. Thus $0 < d < N$.

We will prove by induction that $\sigma_{k'}(j_{k'}) \neq a$ and $\sigma_{k'}(j_k) = a$ holds for all k', $k < k' < k + d$ (i.e. the position of a does not change). Let us have $k < k' < k + d$. We get $\sigma_{k'-1}(j_k) = a$ either by induction assumption, or by the definition of a when $k' = k + 1$. Since $i_{k'} = i_k + (k' - k) \bmod N$ and $1 \leq k' - k < d$, we get $i_{k'} \neq j_k$. Definition 2 implies that $\sigma_{k'}$ differs from $\sigma_{k'-1}$ only at positions $i_{k'}$ and $j_{k'}$ and the monotonous property implies that $i_{k'} \notin \mathrm{Dom}(\sigma_{k'})$. Therefore $\sigma_{k'}(j_{k'}) = \sigma_{k'-1}(i_{k'})$. This value is not equal to a, because $\sigma_{k'-1}(j_k) = a$ and $i_{k'} \neq j_k$. Hence $\sigma_{k'}(j_{k'}) \neq a$ and $\sigma_{k'}(j_k) = \sigma_{k'-1}(j_k) = a$.

Finally we get $\sigma_{k+d}(j_{k+d}) = \sigma_{k+d-1}(i_{k+d}) = \sigma_{k+d-1}(j_k) = a$. □

Lemma 16. *Let T_0 be a reduced monotonous persistent state ($N \in \mathbb{N}$ or $N = \infty$). Then $0 \notin \Sigma$.*

Proof: Assume contrariwise. Since T_0 is reduced, there exists some $k \geq 0$ for which $\sigma_{k+1}(j_{k+1}) = \sigma_k(i_{k+1}) = 0$. So $j_{k+1} = j_k + \sigma_k(i_{k+1}) \bmod N = j_k + 0 \bmod N$. Since $j_k \in \mathrm{Dom}(\sigma_k)$ for all $k \geq 0$ and $j_{k+1} = j_k$, we get $\sigma_{k+1}(i_{k+1}) = \sigma_k(j_{k+1}) \in \Sigma$ (see Definition 2), a contradiction to the monotonous property of T_0. □

Now we will describe the function η for all $a \in \Sigma$ and all $k \geq 1$.

Proposition 17. *Let $T_0 = (i_0, j_0, \sigma_0)$ be a reduced monotonous persistent state ($N \in \mathbb{N}$ or $N = \infty$). Then for all $a \in \Sigma$ and all $k \geq 1$*

$$\begin{array}{ll} \eta_{k+1}(a) = \eta_k(a) - 1, & if\ \eta_k(a) > 1 \\ \eta_{k+1}(a) = \eta_k(\sigma_k(j_k)) + a - 1\ mod\ N, & if\ \eta_k(a) = 1. \end{array}$$

Proof: The first equation follows directly from the definition of η. To prove the second one, assume $\eta_k(a) = 1$. We have $\sigma_{k+1}(j_{k+1}) = a$, which is the same as $\sigma_k(i_{k+1}) = a$. According to Lemma 15 we get $\eta_{k+1}(\sigma_{k+1}(j_{k+1})) = d$, where $0 < d < N$ and $d = j_{k+1} - i_{k+1} \bmod N$. Since $j_{k+1} = j_k + \sigma_k(i_{k+1}) \bmod N$, we also have $d = j_k + \sigma_k(i_{k+1}) - i_{k+1} \bmod N$. Consequently $d = (j_k - i_k) + a - 1 \bmod N = \eta_k(\sigma_k(j_k)) + (a - 1) \bmod N$. □

Notice that only the values $\eta_k(a)$, a and $\eta_k(\sigma_k(j_k))$ are needed to determine the value $\eta_{k+1}(a)$. Thus to determine the function $\eta_{k+1} : \Sigma \to \mathbb{N}^+$ we need to know (in addition to η_k) the value $b \in \Sigma$ such that $b = \sigma_k(j_k)$.

The following definition is stated here, while it is closely related to the notion of the RC4 state, although we will use it in the next section.

Definition 18. *Let $N \in \mathbb{N}$ or $N = \infty$, let $T_0 = (i_0, j_0, \sigma_0)$ be a state of height h and let $j_0 \in Dom(\sigma_0)$. Define a function $\nu : \{0, \ldots, h\} \to \Sigma$ by $\nu(k) = \sigma_k(j_k)$.*

Assume (i, j, S), $S \in \mathcal{S}_N$ is a full RC4 state. In Section 4.3 we will prove (Proposition 30) that every reduced monotonous persistent state (i_0, j_0, σ_0) is periodic, i.e. there exist an integer $l \in \mathbb{N}$ such that $(i_0, j_0, \sigma_0) = (i_l, j_l, \sigma_l)$. At the same time the transition function in the PRGA, which changes (i, j, σ) into $(i' = i + 1 \bmod N,\ j' = j + S(i) \bmod N,\ S' = S \circ (i', j'))$, is invertible.

Since the PRGA sets $i = 0$ and $j = 0$ at the beginning, the periodicity and the invertibility implies that the PRGA can never reach any full RC4 state (i, j, S) which would be an extension of a reduced monotonous persistent k-state ($k << N$). Thus if the PRGA ever reaches an extension of a persistent state, this will not be monotonous. In spite of the fact that monotonous persistent states are not reachable by the PRGA, they can be used for cryptanalysis of RC4 as described in Section 5.4.

Due to the simplicity of the case $N = \infty$, we tend to think that the study of persistent states for $N = \infty$ could serve as a good foundation for studying persistent states generally and so this theory could have further applications in the future. Proposition 14 states that for $N = \infty$ each reduced persistent state is monotonous. Thus we will focus on monotonous states and thereby cover all persistent states for $N = \infty$ and all monotonous persistent states for $N < \infty$.

4.3 The Tabular model

In this section we will introduce the *tabular model* for reduced monotonous persistent states and monotonous profitable states. The model seems to provide a useful environment for description and research of these states.

Firstly we define a *table triple*, which is a mathematical description for the elements of our *tabular model*, and give some examples. Afterwards we will state and prove a proposition about the close relationship of reduced monotonous persistent states and table triples. The rest of the section is devoted to persistent table triples.

Notation: In the following sections Δ_n will denote the set $\{1, \ldots, n\}$ for an $n \in \mathbb{N}$.

Definition 19. *Let $N \in \mathbb{N}$ or $N = \infty$ and $n \in \mathbb{N}$, $n < N$. Let $v : \Delta_n \to \mathbb{Z} \setminus \{-1\}$, $\mu_0 \in \Delta_n$ and $t_0 : \Delta_n \to \mathbb{Z}_N$. If $N = \infty$, assume $t_0(j) > 0$ for all $j \in \Delta_n$. The triple $(v, t_0, \mu_0)_N^n$ will be called a* table triple.

Let $i \geq 0$ and assume that the value $\mu_i \in \Delta_n$ and the mapping $t_i : \Delta_n \to \mathbb{Z}_N$ are defined. Also assume there exists only one $a \in \Delta_n$ such that $t_i(a) = 1$, and that $t_i(k) > 0$ for all $k \in \Delta_n$ if $N = \infty$. Define the value μ_{i+1} and the mapping t_{i+1} in the following way:

$$\mu_{i+1} = a \ , \quad if \ t_i(a) = 1, \ and$$

$$\begin{aligned} t_{i+1}(a) &= t_i(a) - 1 \ mod \ N, && if \ t_i(a) > 1 \ or \ t_i(a) = 0 \\ t_{i+1}(a) &= t_i(\mu_i) + v(a) \ mod \ N, && if \ t_i(a) = 1 \end{aligned}$$

Definition 20. *Let $N \in \mathbb{N}$ or $N = \infty$ and let $(v, t_0, \mu_0)_N^n$ be a table triple. Assume h is the biggest positive integer for which we are able to define μ_h and t_h according to Definition 19. Then we say the triple $(v, t_0, \mu_0)_N^n$ has height h or is* h-profitable. *If such an integer does not exist, we say the table triple $(v, t_0, \mu_0)_N^n$ is* persistent *or that it has height $h = \infty$.*

Remark 21. *The coincidence of above defined terms and terms in Definition 2 is intentional.*

In the rest of this section let us have $N \in \mathbb{N}$ or $N = \infty$, and $n \in \mathbb{N},\ n < N$, if not stated differently.

Lemma 22. *Let $(v, t_0, \mu_0)_N^n$ be a table triple of height h, $h \in \mathbb{N}$. Let $0 \leq k \leq h$ be the smallest integer such that t_k is not injective, thus $t_k(a) = t_k(b)$ for some $a, b \in \Delta_n,\ a \neq b$. Then $h \leq k + t_k(a) - 1$ and t_m (if defined) is not injective for all $m,\ k \leq m \leq h$.*

Proof: Assume $t_k(a) = t_k(b),\ a \neq b$ and set $l = k + t_k(a) - 1$. If t_l is not defined, the inequality $h \leq k + t_k(a) - 1$ holds. If t_l is defined, we get $t_l(a) = t_l(b) = 1$. So we are not able to define a mapping t_l and thus $h \leq l$. The second statement is trivial. □

Lemma 23. *Let $(v, t_0, \mu_0)_N^n$ be a table triple of height h and assume that for some k all $t_i,\ 0 \leq i < k \leq h$, are injective. Then for $0 \leq i \leq k - 1$ the mapping $t_{i+1} : \Delta_n \to \mathbb{Z}_N$ and the value μ_{i+1} uniquely determine the mapping t_i and the value $\mu_i \in \Delta_n$.*

Proof: Clearly $t_i(a) = t_{i+1}(a) + 1 \mod N$ for all $a \in \Delta_n,\ a \neq \mu_{i+1}$ and $t_i(\mu_{i+1}) = 1$. Since $t_{i+1}(\mu_{i+1}) = t_i(\mu_i) + v(\mu_{i+1}) \mod N$, the values $t_{i+1}(\mu_{i+1})$ and $v(\mu_{i+1})$ uniquely determine the value $t_i(\mu_i)$. This determines the value μ_i because we have assumed the injectivity of $t_i,\ 0 < i < k$. Finally, for all $b \neq \mu_{i+1}$ we have $t_i(b) = t_{i+1}(b) + 1 \mod N$. □

We will now explain where the adjective table in the term table triple comes from.

Let us we have a table triple $(v, t_0, \mu_0)_N^n$ of height h. Start with an empty table with 2 rows and n columns. Write values $v(i),\ 1 \leq i \leq n$ in the first row in such a way that $v(i)$ is in the i-th column. In the same manner, write values $t_0(i)$ in the second row. From now on, we will regard the first row of this table as the header of the table and the second row as the zero row (it contains values $t_0(i)$, so the terminology is straight). The value in the first row and the μ_0-th column is highlighted.

Assume $i \geq 0$ and assume that we have already defined the ith row in the table. If $i < h$, we construct the $(i+1)$th row of the table by writing the value $t_{i+1}(j)$ in the j-th column by highlighting the value in the row μ_{i+1} (see Figure 4).

$v(1)$	$v(2)$	$v(3)$
$t_0(1) = 1$	$t_0(2)$	$\mathbf{t_0(3)}$
$\mathbf{t_0(3)} + \mathbf{v(1)}$	$t_0 2 - 1$	$t_0(3) - 1$

Figure 4: A table given by $(v, t_0, \mu_0)_N^3$ with $\mu_0 = 3,\ t_0(1) = 1$ and height $h \geq 1$

It is natural to write mappings from the set Δ_n as n-tuples and we shall adapt this convention. I.e. for $v : \Delta_n \to X$, $v = (x_1, \ldots, x_n)$ will denote the fact that $v(i) = x_i$. We will sometimes write v_i instead of $v(i)$.

Example 24. Let $N \geq 7$. Figure 5 shows a table determined by a table triple $(v, t_0, \mu_0)_N^4$ of height $h = 8$ and $v = (2, -3, 4, 0)$, $t_0 = (1, 2, 3, 5)$ and $\mu_0 = 3$.

2	−3	4	0
1	2	**3**	5
5	1	2	4
4	**2**	1	3
3	1	**6**	2
2	**3**	5	1
1	2	4	**3**
5	1	3	2
4	**2**	2	1
3	1	1	**2**

Figure 5: A table determined by the table triple from Example 24

Example 25. Let $N \geq 20$. Figure 6 shows a table determined by a table triple $(v, t_0, \mu_0)_N^7$ of height $h = 20$ with $v = (-9, 10, -3, 5, 3, 9, 12)$, $t_0 = (1, 2, 11, 4, 5, 7, 9)$ and $\mu_0 = 3$.

Proposition 26. *Assume $N \in \mathbb{N}$ or $N = \infty$. Let $(v, t_0, \mu_0)_N^n$ be a table triple of height h, $h \in \mathbb{N}$ or $h = \infty$. Assume that mappings v and $t_i, 0 \leq i \leq h$ are all injective. Then there exists an n-state $T = (i, j, \sigma)$ of height at least h. Vice versa, if there exists a reduced monotonous persistent n-state, then there exists a uniquely defined persistent table triple $(v, t_0, \mu_0)_N^n$ with an injective mapping v.*

Proof: Let $(v, t_0, \mu_0)_N^n$ be a table triple of height h, i.e. there exist uniquely defined mappings $t_i : \Delta_n \to \mathbb{Z}_N$ (we also assume injectivity) and values μ_i for all $0 \leq i \leq h$. Define an n-state $T_0 = (i_0, j_0, \sigma_0)$ in the following way: Set i_0 as an arbitrary number from $\mathbb{Z}_N$ and set $j_0 = i_0 + \mu_0 \bmod N$. Set $\alpha_0 = \{(i_0 + t_0(k)) \bmod N \mid 1 \leq k \leq n\}$. Define the mapping $\sigma_0 : \alpha_0 \to \mathbb{Z}_N$, such that $\sigma_0(i_0 + t_0(k)) = v(k) + 1 \bmod N$. The injectivity of mappings v and t_0 implies that $|\alpha_0| = n$ and that σ_0 is injective. So $T_0 = (i_0, j_0, \sigma_0)$ is an n-state.

Now we will show that $T_0 = (i_0, j_0, \sigma_0)$ has height at least h. Assume that $0 \leq l \leq h - 1$ and that T_l is already defined. Set $i_{l+1} = i_l + 1 \bmod N$. Because $(v, t_0, \mu_0)_N^n$ has height h, there exists just one $k \in \Delta_n$ such that $t_l(k) = 1$ (otherwise t_{l+1} could not be defined). The definition of α_l implies

−9	10	−3	5	3	9	12
1	2	**11**	4	5	7	9
2	1	10	3	4	6	8
1	**12**	9	2	3	5	7
3	11	8	1	2	4	6
2	10	7	**8**	1	3	5
1	9	6	7	**11**	2	4
2	8	5	6	10	1	3
1	7	4	5	9	**11**	2
2	6	3	4	8	10	1
1	5	2	3	7	9	**14**
5	4	1	2	6	8	13
4	3	**2**	1	5	7	12
3	2	1	**7**	4	6	11
2	1	**4**	6	3	5	10
1	**14**	3	5	2	4	9
5	13	2	4	1	3	8
4	12	1	3	**8**	2	7
3	11	**5**	2	7	1	6
2	10	4	1	6	**14**	5
1	9	3	**19**	5	13	4
10	8	2	18	4	12	3

Figure 6: A table determined the table triple from Example 25

that $i_l + 1 \in \alpha_l$ and thus $i_l + 1 \in \mathrm{Dom}(\sigma_l)$. So we can define the value $j_{l+1} = j_l + \sigma_l(i_l + 1) \bmod N$. Finally set $\alpha_{l+1} = \{(i_{l+1} + t_{l+1}(k)) \bmod N \mid 1 \leq k \leq n\}$ (we get $|\alpha_{l+1}| = n$ out of the injectivity of t_{l+1}) and $\sigma_{l+1} : \alpha_{l+1} \to \mathbb{Z}_N$, $\sigma_{l+1}(i_{l+1} + t_{l+1}(k) \bmod N) = v(k) + 1 \bmod N$. The mapping v is injective, hence σ_{l+1} is a partial permutation of $\mathbb{Z}_N$. Thus the first part of the proposition is proved.

Let $T_0 = (i_0, j_0, \sigma_0)$ be a reduced monotonous persistent n-state. Since σ_0 is injective, $|\mathrm{Dom}(\sigma_0)| = |\Sigma| = n$ (Σ denotes $\mathrm{Im}(\sigma_0)$). Denote elements of Σ as a_i, i.e. $\Sigma = \{a_1, \ldots, a_n\}$.

Define the mapping $v : \Delta_n \to \mathbb{Z}$ by $v(i) = a_i - 1$. Lemma 16 implies that $0 \notin \Sigma$. Thus $v(i) \neq -1$ holds for all $i \in \Delta_n$ and we have $v : \Delta_n \to \mathbb{Z} \setminus \{-1\}$. Because σ_0 is injective (it is a partial permutation), v is also injective. For all $k \geq 0$ define the mapping $t_k : \Delta_n \to \mathbb{Z}_N$ by $t_k(i) = \eta_k(a_i)$ for all $i \in \Delta_n$. Also for all $k \geq 0$ set $\mu_k = i$ if and only if $\nu(k) = a_i$. The requested equations

$$\begin{aligned} t_{i+1}(a) &= t_i(a) - 1 \bmod N, && if\ t_i(a) > 1 \\ t_{i+1}(a) &= v(a) + t_i(\mu_i) \bmod N, && if\ t_i(a) = 1 \end{aligned}$$

and the condition $\mu_{i+1} = a$ for $t_i(a) = 1$ follow from Definition 18 and Proposition 17. If $N = \infty$, the condition $t_i(a) > 0$ for all $a \in \Delta_n$ follows from Proposition 14. □

In the construction of an n-state out of a table triple we have chosen the number i_0. Because there are N possibilities how to choose this number, we see that a profitable table triple determines a set of n-states, parametrised by the choice of i_0.

One of assumptions in Proposition 26 is the injectivity of the mapping $v : \Delta_n \to \mathbb{Z} \setminus \{-1\}$. On the other hand, this property of the mapping v was not assumed in Definition 19 and thus the notion of the table triple is more general than the notion of the reduced monotonous persistent state. The system defined by Definition 19 is mathematicly interesting in itself, so it is worth considering such a generalisation.

From now on we will focus on persistent table triples. To start we will describe the only reduced monotonous persistent state that seem to be known. According to Proposition 26 this state gives rise to a persistent table triple. This state was first described in [Fi94], thus it is called the Finney state.

Example 27. *(Finney state)* Let $N \in \mathbb{N}$ or $N = \infty$ and $k \in \mathbb{Z}_N$. Define a 1-state (i_0, j_0, σ_0) in the following way: $i_0 = k,\ j_0 = i_0 + 1 \bmod N\ = k+1 \bmod N$ and $\sigma_0(j_0) = 1$, $|\Sigma| = 1$. Then $i_1 = k + 1 \bmod N,\ j_1 = j_0 + \sigma_0(i_1) \bmod N = k + 1 + 1 \bmod N = i_1 + 1 \bmod N$ and $\sigma_1(j_1) = 1$. In the same way, we get $i_l = k + l \bmod N,\ j_l = i_l + 1 \bmod N$ and $\sigma_l(j_l) = 1$ for all $l \geq 0$. We see that the state (i_0, j_0, σ_0) is persistent. According to Proposition 26 it gives rise to the persistent table triple described by the table on Figure 7.

0
1
1

Figure 7: A persistent table triple determined by the Finney state

Let $(v, t, \mu)^n_N$ be a table triple of height h. Assume there exist i and j, $0 \leq i < j \leq h$ for which $t_i = t_j$ and $\mu_i = \mu_j$, and let i be the smallest integer of this kind. Then clearly all t_l, $0 \leq l \leq j$ are injective and $(v, t, \mu)^n_N$ is persistent. Assume $i \neq 0$. If follows from Lemma 23 that $t_{i-1} = t_{j-1}$, which is in contradiction with the minimality of i. Thus $i = 0$ and $t_x = t_{x \bmod j}$ for all $x \geq 0$.

If $N < \infty$, then the set $\{(t_i, \mu_i) \mid t_i : \Delta_n \to \mathbb{Z}_N, \mu_i \in \Delta_n\}$ has a finite cardinality. Thus for all $N < \infty$ each persistent table $(v, t, \mu)^n_N$ is periodic. We will now prove that all persistent table triples are periodic even in the case $N = \infty$.

For this purpose we will state an auxiliary definition and we will prove one lemma.

Notation: If $a \in \Delta_n$, $a \neq \mu_0$, then by $(v^{(-a)}, t_0^{(-a)}, \mu_0)_N^n$ we will denote a table triple which arises from restriction of mappings t_0 and v to the set $\Delta_n \setminus \{a\}$.

Definition 28. *A table triple $(v, t_0, \mu_0)_N^n$ is said to be* restriction persistent*, if one of the following conditions holds:*

1. *$(v, t, \mu)_N^n$ is persistent,*
2. *for some $a \in \Delta_n$, $a \neq \mu_0$, $(v^{(-a)}, t_0^{(-a)}, \mu_0)_N^n$ is restriction persistent,*
3. *$a = \mu_0$, $(v, t_0, \mu_0)_N^n$ has height $h \geq 2$, $\mu_1 \neq a$ and $(v^{(-a)}, t_1^{(-a)}, \mu_1)_N^n$ is restriction persistent.*

Lemma 29. *Let $N = \infty$, $s \geq 1$ and $m \geq 1$. Then there exist numbers $f(m, s)$ and $g(m, s)$ such that each table triple $(v, t_0, \mu_0)_N^n$ that is not restriction persistent and satisfies $s \geq n$, $m \geq \max\{|v(a)| \mid a \in \Delta_n\}$ and $f(m, s) \leq \max\{t_0(a) \mid a \in \Delta_n\}$ has height $h \leq g(m, s)$.*

Proof: Set $f(m, 1) = 2$. Clearly we can set $g(m, 1) = 1$. We will continue by induction. The value $f(m, s+1)$ will be set so that $f(m, s+1) > g(m, s)$ and $f(m, s+1) > f(m, s)$. Assume that the lemma holds for s.

Let $(v, t_0, \mu_0)_N^n$ be a table triple with $n = s+1 \geq 2$ and $t_0(a) > \max\{f(m, s), g(m, s) + 1\}$ for some $a \in \Delta_n$ which is not reduction persistent. The rest of the proof is divided into two parts.

a) Assume that there exists $b \in \Delta_n$, $b \neq a$ for which $t_0(b) > f(m, s)$. We can also assume that $(v, t_0, \mu_0)_N^n$ has height at least 2, so $(v, t_1, \mu_1)_N^n$ is defined and $t_1(b) \geq f(m, s)$ holds. If $\mu_0 = a$ then $\mu_1 \neq a$ because $t_0(a) > 2$. If $\mu_0 = a$ set $j = 1$. Otherwise set $j = 0$, so $\mu_j \neq a$ in both cases. We will consider the table triple $(v, t_j, \mu_j)_N^n$ as the initial one. Because $(v^{(-a)}, t_j^{(-a)}, \mu_j)_N^n$ has $n = s$ and $t_j(b) \geq f(m, s)$ we can use the induction assumption. Thus there exists $h \leq g(m, s)$ such that $t_h^{(-a)}$ and $\mu_h^{(-a)}$ are the last which can be defined. Therefore either $t_h^{(-a)}(x) \leq 0$ for some $a \neq x \in \Delta_n$ or $t_h^{(-a)}(x) = t_h^{(-a)}(y)$ for $x \neq y \in \Delta_n$ or $t_h^{(-a)}(x) \neq 1$ for all $a \neq x \in \Delta_n$. Since $t_0(a) > g(m, s) + 1$, this problem is not removed by the value $t_h(a)$ and it is shared by the mapping t_h. So the height of the table triple $(v, t_0, \mu_0)_N^n$ is less or equal than $g(m, s)$.

b) Assume that $t_0(b) \leq f(m, s)$ for all $b \in \Delta_n$, $b \neq a$. Again set $j = 1$ if $\mu_0 = a$, otherwise set $j = 0$. Thus $\mu_j \neq a$. There are finally many table triples $(v^{(-a)}, t_j^{(-a)}, \mu_j)_N^n$ with $m \geq \max\{|v(a)| \mid b \in \Delta_n \setminus \{a\}\}$ and $t_j^{(-a)}(b) \leq f(m, s)$ for all $b \in \Delta_n \setminus \{a\}$. Since $(v, t_0, \mu_0)_N^n$ is not reduction persistent, none of these table triples is persistent. Thus there exists an integer h_{max} such that the height of each of these table triples is less then h_{max}. Consequently if $t_0(a) > \max\{f(m, s), g(m, s) + 1, h_{max}\}$, then $t_h(a) > 1$ for all $0 \leq h \leq h_{max}$, and $t_h(a)$ cannot enlarge the height of any of these table triples.

To finish the proof it is sufficient to set $f(m, s+1) = \max\{f(m,s) + 1, g(m,s) + 2, h_{max} + 1\}$ and $g(m, s+1) = \max\{g(m,s) + 1, h_{max}\}$. □

Proposition 30. *Each persistent table triple $(v, t, \mu)_N^n$ is periodic, i.e. it generates a periodical table.*

Proof: We have already observed that this proposition holds for all $N < \infty$, thus it is enough to consider the case $N = \infty$. We will prove that for any given m and s there exist just finally many table triples $(v, t_0, \mu_0)_N^n$ with $n \leq s$ and $m \geq \max\{|v(a)| \mid a \in \Delta_n\}$ which are persistent. The periodicity then follows from this finiteness property.

To prove that the mentioned set of table triples is finite it suffices to show that if $(v, t_0, \mu_0)_N^n$ is persistent, then for all $a \in \Delta_n$

$$t_0(a) \leq \max\{f(m,s), g(m,s) + 1\}.$$

Assume the opposite and let $(v, t_0, \mu_0)_N^n$ be a persistent triple with an $a \in \Delta_n$ such that $t_0(a) > \max\{f(m,s), g(m,s)+1\}$. Then $\mu_1 \neq a$ and $(v^{(-a)}, t_1^{(-a)}, \mu_1)_N^n$ is not reduction persistent (if it is, we will get the contradiction with the injectivity of the mapping $t_{t_0(a)-1}$ and hence the contradiction with the persistence of $(v, t_0, \mu_0)_N^n$). Also

$$t_1(a) \geq \max\{f(m,s), g(m,s) + 1\}$$

so we can use Lemma 29. Hence h, the height of $(v^{(-a)}, t_1^{(-a)}, \mu_1)_N^n$, is less than or equal to $g(m,s)$. Since $t_h(a) = t_0(a) - h \geq 2$ this value cannot increase the height of this state, and hence $(v, t_0, \mu_0)_N^n$ is not persistent. □

Notation: Let $(v, t, \mu)_N^n$ be a persistent table triple. Then P will denote the period of this table triple, i.e. P is the smallest positive integer for which $t_0 = t_P$ and $\mu_0 = \mu_P$.

Example 31. A table triple determined by the Finney state (Example 27) has period equal to 1.

As already said in this section, the notion of table triple is more general that the notion of the reduced monotonous n-state, because we do not assume the injectivity of the mapping v. We will now show some examples of persistent table triples with a non-injective mapping v.

Example 32. Let $N \geq 5$. Figure 8 shows a persistent table triple $(v, t_0, \mu_0)_N^3$ with $v = (-2, 2, 2),\ t_0 = (1, 2, 4),\ \mu_0 = 3$. This table triple has a period equal to 4.

Example 33. . Let $N \geq 7$. Figure 9 shows a persistent table triple $(v, t_0, \mu_0)_N^4$ with $v = (-3, 0, 3, 3),\ t_0 = (1, 2, 3, 6),\ \mu_0 = 4$. This table triple has a period equal to 6.

-2	2	2
1	2	**4**
2	1	3
1	**4**	2
2	3	1
1	2	**4**

Figure 8: A persistent table with non-injective mapping v (Example 32).

-3	0	3	3
1	2	3	**6**
3	1	2	5
2	**3**	1	4
1	2	**6**	3
3	1	5	2
2	**3**	4	1
1	2	3	**6**

Figure 9: A persistent table with non-injective mapping v (Example 33)

5 Quest for persistent states

We will start this section with a description of some further properties of persistent table triples, which will help us to prove the main results of this work. In this section we will denote the set $\{1, \ldots, n\}$ by Δ_n, as we have done in Section 4.3.

Lemma 34. *Let $(v, t_0, \mu_0)_N^n$ be a persistent table triple. Then for all $i \in \Delta_n$ there exists $j \geq 1$, such that $\mu_j = i$.*

Proof: Let $i \in \Delta_n$ and $k = t_0(i)$. Then $t_{k-1}(i) = 1$ and thus $\mu_k = i$. □

Definition 35. *Let $(v, t, \mu)_N^n$ be a persistent table triple with period P. Define the mapping $\mu : \Delta_P \to \Delta_n$ as $\mu(i) = \mu_i$.*

Definition 36. *Let $(v, t, \mu)_N^n$ be a persistent table triple with period P. For all $i \in \Delta_n$, define m_i as the number of all integers $j \in \Delta_P$ for which $\mu(j) = i$. Thus*

$$m_i = |\mu^{-1}(i)|.$$

Lemma 37. *Let $(v, t, \mu)_N^n$ be a persistent table triple with period P. Then*

$$P = \sum_{i=1}^{n} m_i.$$

Proof: $\{1, \ldots, P\} = \mu^{-1}(1) \cup \mu^{-1}(2) \cup \ldots \cup \mu^{-1}(n)$ and $\mu^{-1}(i) \cap \mu^{-1}(j) = \emptyset$ for all $i \neq j$. □

Lemma 38. *Let $(v, t_0, \mu_0)_N^n$ be a persistent table triple with period P. Then for all $k \in \Delta_P$*

$$t_k(\mu(k)) = t_0(\mu_0) + \sum_{j=1}^{k} v(\mu(j)).$$

Proof: For $k = 0$ the statement is trivial. Assume that the lemma holds for $k \geq 0$, so $t_k(\mu(k)) = s + \sum_{j=1}^{k} v(\mu(j))$. Definition 19 directly implies that $t_{k+1}(\mu(k+1)) = t_k(\mu(k)) + v(\mu(k+1))$. This proves the lemma. □

Lemma 38 also holds for $k = P$, therefore $t_P(\mu(P)) = t_0(\mu_0) + \sum_{j=1}^{P} v(\mu(j))$. On the other hand the periodicity of the table triple $(v, t_0, \mu_0)_N^n$ gives us $t_P(\mu(P)) = t_0(\mu(0)) = t_0(\mu_0)$. We have proven

Corollary 39. *Let $(v,t,\mu)_N^n$ be a persistent table triple with period P. Then*

$$\sum_{j=1}^{P} v(\mu(j)) = 0.$$

Directly from Corollary 39 and Lemma 37 we obtain

Corollary 40. *Let $(v,t,\mu)_N^n$ be a persistent table triple with period P. Then*

$$\sum_{i=1}^{n} m_i \cdot v(i) = 0. \tag{1}$$

Proposition 41. *Let $(v,t,\mu)_N^n$ be a persistent table triple with period P. Then for all $i \in \Delta_n$*

$$P = \sum_{j \in \mu^{-1}(i)} t_j(i).$$

Proof: Let $i \in \Delta_n$ and $j \in \mu^{-1}(i)$. Set $k = t_j(i)$. We will consider two cases. Firstly assume that $j + k - 1 \leq P$. Then for all l, $j + 1 \leq l \leq j + k - 1$ we have $t_l(i) = t_{l-1}(i) - 1 = k - (l - j)$ since $t_{l-1}(i) \neq 1$. Consequently $\mu(l) \neq i$ for all $j + 1 \leq l \leq j + k - 1$. Thus $t_{j+k-1}(i) = 1$ and $\mu(j + k) = i$. We see that the value k determines the length of the decreasing sequence in the column i, which begins with the value $t_j(i)$ (in the j-th row) and ends with the value 1 (in the row $(j + k - 1)$).

Consider the second case, $j + k - 1 > P$. The periodicity of the table gives us the identity of mappings t_i and $t_{i \bmod P}$. So we can use the first case if we consider the mapping $t_{l \bmod P}$ instead of t_l. In the same manner we obtain that $t_j(i)$ is the length of the decreasing sequence in the column i which ranges from the row j to the row P and then continues from the row 1 to the row $j + k - 1 \bmod P$.

There are m_i such decreasing sequences in the column i and rows $1, \ldots, P$ and all their lengths are equal to values $t_j(i)$ for $j \in \mu^{-1}(i)$. Thus the proposition is proved.

□

Using the previous proposition and Lemma 38 we directly obtain

Corollary 42. *Let $(v,t_0,\mu_0)_N^n$ be a persistent table triple with period P. Then for all $i \in \Delta_n$*

$$P = t_0(\mu_0)m_i + \sum_{k \in \mu^{-1}(i)} \sum_{l=1}^{k} v(\mu(l)). \tag{2}$$

The mapping v has $\mathrm{Im}(v) = \Delta_n = \{1, \ldots, n\}$ as its image set. Hence we can reorder the double sum in equation (2) and gather all summands $v(l)$ with the same $l \in \Delta_n$. In this way we get numbers $\lambda_{i,j}$.

Definition 43. *Let $(v,t,\mu)_N^n$ be a persistent table triple with period P. For all $i,j \in \Delta_n$ define an integer $\lambda_{i,j} \in \mathbb{N}$ as the number of pairs (k,l) such that $\mu(l) = j$ and $l \leq k \in \mu^{-1}(i)$. I.e.*

$$\lambda_{i,j} = |\{(k,l) \in \mu^{-1}(i) \times \mu^{-1}(j) \mid l \leq k\}|.$$

Lemma 44. *Let $(v,t,\mu)_N^n$ be a persistent table triple with period P. Then for all $i \in \Delta_n$*

$$P = t_0(\mu_0)m_i + \sum_{j=1}^{n} \lambda_{i,j}\ v(j). \tag{3}$$

Proof: We will prove the lemma using Corollary 42. Let $\mu^{-1}(i)$ be equal to the set $\{a_1, a_2, \ldots, a_{m_i}\}$ where $a_i < a_{i+1}$. Assume that the value $v(j)$ appears in the sum $\sum_{l=1}^{a_k} v(\mu(l))$ from equation (2) for some $k, 1 \leq k \leq m_i$. Because this sum starts always with $l = 1$, the value $v(j)$ will appear in each of the following sum $\sum_{l=1}^{a_r} v(\mu(l))$ for all $k \leq r \leq m_i$. Thus $v(j)$ is contained in the sum $\sum_{l=1}^{a_k} v(\mu(l))$ so many times, how many $u < a_k$ there are with the property $\mu(u) = j$. □

5.1 Matrices induced by equivalences

In the beginning of this section we will show a relationship between table triples and equivalences on an ordered set. Then we will define a matrix Λ induced by an equivalence and prove that this matrix is regular for any equivalence.

Definition 45. *Let $\sim$ be an equivalence on a linearly ordered set $A = \{a_1, \ldots, a_P\}$ with n equivalence classes, $n \geq 1$. Denote these classes by $C_1, \ldots, C_n$ and denote the linear ordering by $\leq$. Set $m_i = |C_i|$. For each $i,j \in \Delta_n$ define a value $\lambda_{i,j}$*

$$\lambda'_{i,j} = |\{(a_k, a_l) \in C_i \times C_j \mid a_k \geq a_l\}|.$$

Assume that $T_0 = (v, t_0, \mu_0)_N^n$ is a persistent table triple with period P (we assume $N \in \mathbb{N}$ or $N = \infty$). The mapping $\mu : \Delta_P \to \Delta_n$ from Definition 35 defines a relation $\sim$ on the set Δ_P such that $i \sim j$ if $\mu(i) = \mu(j)$. Clearly $\sim$ is an equivalence on Δ_P and Lemma 34 implies that $\sim$ has n equivalence classes. Set $C_i = \mu^{-1}(i),\ 1 \leq i \leq n$. Since we have the less or equal ordering on the set Δ_P, we can define values $\lambda'_{i,j}$ for all $i,j \in \Delta_n$. Definition 43 implies that for a persistent table triple T_0 and the equivalence $\sim$ determined by this table triple we obtain $\lambda_{i,j} = \lambda'_{i,j}$. Thus we can write $\lambda_{i,j}$ also for equivalences.

Lemma 46. *Let $\sim$ be an equivalence on a linearly ordered set $\{a_1, \ldots, a_P\}$ with n equivalence classes, $n \geq 2$. Then for all $i,j \in \Delta_n$, $i \neq j$*

$$\lambda_{i,j} + \lambda_{j,i} = m_i m_j. \tag{4}$$

Proof: This follows directly from Definition 45. □

Lemma 47. *Let $\sim$ be an equivalence on a linearly ordered set $\{a_1, \ldots, a_P\}$ with n equivalence classes, $n \geq 2$. Then for all $i \in \Delta_n$*

$$\lambda_{i,i} = \frac{m_i(m_i+1)}{2}. \tag{5}$$

Proof: This follows directly from Definition 45. □

Definition 48. *Let $\sim$ be an equivalence on a linearly ordered set $\{a_1, \ldots, a_P\}$ with n equivalence classes, $n \geq 1$. Define a matrix $\Lambda_n \in (\mathbb{R})_{(n+1)\times(n+1)}$ by*

$$\Lambda_n = \begin{pmatrix} 0 & m_1 & m_2 & \ldots & \ldots & m_n \\ m_1 & \lambda_{1,1} & \lambda_{1,2} & \ldots & \ldots & \lambda_{1,n} \\ m_2 & \lambda_{2,1} & \lambda_{2,2} & \ldots & \ldots & \lambda_{2,n} \\ \vdots & \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ m_n & \lambda_{n,1} & \lambda_{n,2} & \ldots & \ldots & \lambda_{n,n} \end{pmatrix}$$

Theorem 49. *Let $\sim$ be an equivalence on a linearly ordered set $\{a_1, \ldots, a_P\}$ with n equivalence classes, $n \geq 1$. Then the matrix Λ_n is regular.*

Proof:

We will prove the theorem by induction in n.

For $n = 1$ we have $\Lambda_1 = \begin{pmatrix} 0 & m_1 \\ m_1 & \lambda_{1,1} \end{pmatrix}$, and Λ_1 is regular since $\det \Lambda_1 = -m_1^2 < 0$.

Let $n \geq 2$. Denote the first row of the matrix Λ_n by $\mathbf{v_0}$, so $\mathbf{v_0} = (0,\ m_1,\ \ldots,\ m_n)$ and denote the row $(i+1)$ (for $0 \leq i < n$) by $\mathbf{v_i}$, so $\mathbf{v_i} = (m_i,\ \lambda_{i,1},\ \ldots,\ \lambda_{i,n})$.

Lemma 50. *Assume that there exist $c_0, \ldots, c_{n-1} \in \mathbb{R}$, not all equal to 0, such that*

$$c_0 \cdot \mathbf{v_0} + c_1 \cdot \mathbf{v_1} + \ldots + c_{n-1} \cdot \mathbf{v_{n-1}} = \mathbf{v_n}. \tag{6}$$

Then

$$0 = \sum_{i=1}^{n-1} c_i^2 m_i + \sum_{i=1}^{n-1} c_i m_i.$$

Proof:

Vector equation (6) is indeed $n+1$ equations in $\mathbb{R}$. We will now look on these equations in $\mathbb{R}$ one after another.

1) Equation (6) for the first coordinate:

$$c_1 m_1 + c_2 m_2 + \ldots + c_{n-1} m_{n-1} = m_n. \tag{7}$$

2) Equation (6) for the second coordinate:

$$c_0 m_1 + c_1 \lambda_{1,1} + c_2 \lambda_{2,1} + \ldots + c_{n-1} \lambda_{n-1,1} = \lambda_{n,1}.$$

Using equations (5) and (4) we obtain

$$c_0 m_1 + c_1 \frac{1}{2} m_1 (m_1 + 1) + c_2 (m_1 m_2 - \lambda_{1,2}) + \ldots$$

$$\ldots + c_{n-1} (m_1 m_{n-1} - \lambda_{1,n-1}) = m_1 m_n - \lambda_{1,n}.$$

Using equation (7) for the right-hand side we get

$$c_0 m_1 + \frac{1}{2} c_1 m_1^2 + \frac{1}{2} c_1 m_1 + \sum_{i=2}^{n-1} (c_i m_1 m_i - c_i \lambda_{1,i}) =$$

$$= m_1 \cdot (\sum_{i=1}^{n-1} c_i m_i) - \lambda_{1,n}$$

and so

$$\lambda_{1,n} = -c_0 m_1 + \frac{1}{2} c_1 m_1^2 - \frac{1}{2} c_1 m_1 + \sum_{i=2}^{n-1} c_i \lambda_{1,i}. \tag{8}$$

3) Equation (6) for the third coordinate:

$$c_0 m_2 + c_1 \lambda_{1,2} + c_2 \lambda_{2,2} + \ldots + c_{n-1} \lambda_{n-1,2} = \lambda_{n,2}.$$

Using equations (5) and (4) we obtain

$$c_0 m_2 + c_1 \lambda_{1,2} + \frac{1}{2} c_2 m_2^2 + \frac{1}{2} c_2 m_2 + \sum_{i=3}^{n-1} (c_i m_2 m_i - c_i \lambda_{2,i}) =$$

$$= m_2 \cdot (\sum_{i=1}^{n-1} c_i m_i) - \lambda_{2,n}$$

and so

$$\lambda_{2,n} = -c_0 m_2 - c_1 \lambda_{1,2} + \frac{1}{2} c_2 m_2^2 - \frac{1}{2} c_2 m_2 + c_1 m_1 m_2 + \sum_{i=3}^{n-1} c_i \lambda_{2,i}. \tag{9}$$

4) Equation (6) for the fourth coordinate:

$$c_0 m_3 + c_1\lambda_{1,3} + c_2\lambda_{2,3} + \ldots + c_{n-1}\lambda_{n-1,3} = \lambda_{n,3}.$$

Using equations (5) and (4) we obtain

$$c_0 m_3 + c_1\lambda_{1,3} + c_2\lambda_{2,3} + \frac{1}{2}c_3 m_3^2 + \frac{1}{2}c_3 m_3 + \sum_{i=4}^{n-1}(c_i m_3 m_i - c_i\lambda_{3,i}) =$$

$$= m_3 \cdot (\sum_{i=1}^{n-1} c_i m_i) - \lambda_{3,n}$$

and so

$$\lambda_{3,n} = -c_0 m_3 - c_1\lambda_{1,3} - c_2\lambda_{2,3} + \frac{1}{2}c_3 m_3^2 - \frac{1}{2}c_3 m_3 +$$

$$+c_1 m_1 m_3 + c_2 m_2 m_3 + \sum_{i=4}^{n-1} c_i\lambda_{4,i}. \tag{10}$$

So generally for $0 < i < n-1$:

i+1) Equation (6) for the coordinate $(i+1)$:

$$c_0 m_i + c_1\lambda_{1,i} + c_2\lambda_{2,i} + \ldots + c_{n-1}\lambda_{n-1,i} = \lambda_{n,i},$$

$$c_0 m_i + \sum_{j=1}^{i-1} c_j\lambda_{j,i} + \frac{1}{2}c_i m_1^2 + \frac{1}{2}c_1 m_1 + \sum_{j=i+1}^{n-1}(c_j m_i m_j - c_j\lambda_{i,j}) =$$

$$= m_i \cdot (\sum_{j=1}^{n-1} c_j m_j) - \lambda_{i,n},$$

$$c_0 m_i + \sum_{j=1}^{i-1} c_j\lambda_{j,i} + \frac{1}{2}c_i m_1^2 + \frac{1}{2}c_1 m_1 - \sum_{j=i+1}^{n-1} c_j\lambda_{i,j} = m_i \cdot (\sum_{j=1}^{i-1} c_j m_j) + c_i m_i^2 - \lambda_{i,n}$$

and so

$$\lambda_{i,n} = -c_0 m_i - \sum_{j=1}^{i-1} c_j\lambda_{j,i} + \frac{1}{2}c_i m_i^2 - \frac{1}{2}c_i m_i + \sum_{j=1}^{i-1} c_j m_i m_j + \sum_{j=i+1}^{n-1} c_j\lambda_{i,j}. \tag{11}$$

Finally the equation for the last coordinate, i.e. for the last column of the matrix:

$$c_0 m_n + c_1\lambda_{1,n} + c_2\lambda_{2,n} + \ldots + c_{n-1}\lambda_{n-1,n} = \lambda_{n,n}.$$

Using equations (7) and (11) we obtain

$$c_0 \cdot \sum_{j=1}^{n-1} c_j m_j + c_1(-c_0 m_1 \frac{1}{2} c_1 m_1^2 - \frac{1}{2} c_1 m_1 + \sum_{i=2}^{n-1} c_i \lambda_{1,i}) + \dots$$

$$\dots + c_i(-c_0 m_i - \sum_{j=1}^{i-1} c_j \lambda_{j,i} + \frac{1}{2} c_i m_i^2 - \frac{1}{2} c_i m_i + \sum_{j=1}^{i-1} c_j m_i m_j + \sum_{j=i+1}^{n-1} c_j \lambda_{i,j}) + \dots =$$

$$= \frac{1}{2} m_n(m_n + 1) = \frac{1}{2}(\sum_{i=1}^{n-1} c_i m_i)(\sum_{i=1}^{n-1} c_i m_i + 1),$$

$$\sum_{i=1}^{n-1} c_i \cdot (-c_0 m_i - \sum_{j=1}^{i-1} c_j \lambda_{j,i} + \frac{1}{2} c_i m_i^2 - \frac{1}{2} c_i m_i + \sum_{j=1}^{i-1} c_j m_i m_j + \sum_{j=i+1}^{n-1} c_j \lambda_{i,j}) +$$

$$+ c_0 \cdot \sum_{j=1}^{n-1} c_j m_j = \frac{1}{2}(\sum_{i=1}^{n-1} c_i m_i + \sum_{i=1}^{n-1} c_i^2 m_i^2 + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n-1} 2 c_i c_j m_i m_j),$$

$$c_0 \sum_{j=1}^{n-1} c_j m_j - \sum_{i=1}^{n-1} c_0 c_i m_i - \sum_{i=1}^{n-1} \sum_{j=1}^{i-1} c_i c_j \lambda_{j,i} + \frac{1}{2} \sum_{i=1}^{n-1} c_i^2 m_i^2 - \frac{1}{2} \sum_{i=1}^{n-1} c_i^2 m_i +$$

$$+ \sum_{i=1}^{n-1} \sum_{j=1}^{i-1} c_i c_j m_i m_j + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n-1} c_i c_j \lambda_{i,j} =$$

$$= \frac{1}{2} \sum_{i=1}^{n-1} c_i m_i + \frac{1}{2} \sum_{i=1}^{n-1} c_i^2 m_i^2 + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n-1} c_i c_j m_i m_j,$$

$$\underbrace{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n-1} c_i c_j \lambda_{i,j} - \sum_{i=1}^{n-1} \sum_{j=1}^{i-1} c_i c_j \lambda_{j,i}}_{=0} + \underbrace{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n-1} c_i c_j m_i m_j - \sum_{i=1}^{n-1} \sum_{j=1}^{i-1} c_i c_j m_i m_j}_{=0} =$$

$$= \frac{1}{2}(\sum_{i=1}^{n-1} c_i^2 m_i + \sum_{i=1}^{n-1} c_i m_i).$$

Note that we have used the following equalities:

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n-1} c_i c_j \lambda_{i,j} = \sum_{i=1}^{n-1} \sum_{i<j} c_i c_j \lambda_{i,j} = \sum_{i=1}^{n-1} \sum_{j<i}^{n-1} c_i c_j \lambda_{j,i} = \sum_{i=1}^{n-1} \sum_{j=1}^{i-1} c_i c_j \lambda_{j,i}.$$

Thus we have shown that if equation (6) holds then

$$0 = \sum_{i=1}^{n-1} c_i^2 m_i + \sum_{i=1}^{n-1} c_i m_i. \tag{12}$$

□

We will now continue with the proof of the theorem. Let $n \geq 2$ and let $\sim$ be an equivalence on a linearly ordered set $A = \{a_1, \ldots, a_P\}$ with equivalence classes $C_1, \ldots, C_n$. Assume that the theorem holds for $n-1$ and that the matrix Λ_n determined by $\sim$ is singular. Thus rows of the matrix Λ_n are linearly dependent and there exist $a_0, \ldots, a_n \in \mathbb{R}$, not all equal to 0, such that

$$a_0 \cdot \mathbf{v_0} + a_1 \cdot \mathbf{v_1} + \ldots + a_{n-1} \cdot \mathbf{v_{n-1}} + a_n \cdot \mathbf{v_n} = \mathbf{0}. \tag{13}$$

Assume $a_i = 0$ for some $i \in \Delta_n$. Set $B = A \setminus C_i$ and denote by $\sim_B$ the restriction of $\sim$ to the set B. Then the matrix Λ_n without the $(i+1)$-th row and the $(i+1)$-th column is a matrix determined by the equivalence $\sim_B$. Since then equation (13) contradicts the induction assumption, we can assume $a_i \neq 0$ for all $i \in \Delta_n$.

Divide equation (13) by a_n. We obtain

$$-\frac{a_0}{a_n} \cdot \mathbf{v_0} - \frac{a_1}{a_n} \cdot \mathbf{v_1} - \ldots - \frac{a_{n-1}}{a_n} \cdot \mathbf{v_{n-1}} = \mathbf{v_n}.$$

Set $c_i = -\frac{a_i}{a_n}$. We have obtained the equation

$$c_0 \cdot \mathbf{v_0} + c_1 \cdot \mathbf{v_1} + \ldots + c_{n-1} \cdot \mathbf{v_{n-1}} = \mathbf{v_n}$$

and Lemma 50 implies

$$0 = \sum_{i=1}^{n-1} c_i^2 m_i + \sum_{i=1}^{n-1} c_i m_i.$$

Using equation (7) and the fact that $m_i = |C_i| > 0$ for all $i \in \Delta_n$ we get

$$0 = \sum_{i=1}^{n-1} c_i^2 m_i + \sum_{i=1}^{n-1} c_i m_i = \sum_{i=1}^{n-1} \underbrace{c_i^2 m_i}_{>0} + \underbrace{m_n}_{>0} > 0.$$

Assuming the singularity of Λ_n yields a contradiction, thus Λ_n is regular.

□

Proposition 51. *Let $\sim$ be an equivalence on a linearly ordered set $A = \{a_1, \ldots, a_P\}$ with n equivalence classes $C_1, \ldots, C_n$, $n \geq 2$ and let Λ_n be a matrix determined by this equation. Then for all $2 \leq k \leq n$ each principal minor $M_k \in (\mathbb{Z})_{k\times k}$ of Λ_n is regular.*

Proof: Let $2 \leq k \leq n$. Set $B = A\backslash(C_k \cup C_{k+1} \cup \ldots \cup C_n)$ and set $\sim_B$ a reduction of $\sim$ to the set B. Then clearly M_k is a matrix determined by the equivalence $\sim_B$. Thus we can use the Theorem 49. □

At the end of this section we will explicitly state solutions of a system of linear equations determined by a matrix Λ_n and the right-hand side vector $(0, \sum_{i=1}^{n} m_i, \ldots, \sum_{i=1}^{n} m_i)^T \in \mathbb{R}^{n+1}$ for $n = 2$ and $n = 3$.

Consider the following system of linear equations (x_0, x_1, x_2 are the unknown variables)

$$\begin{pmatrix} 0 & m_1 & m_2 \\ m_1 & \frac{1}{2}m_1(m_1+1) & \lambda_{1,2} \\ m_2 & m_1m_2 - \lambda_{1,2} & \frac{1}{2}m_2(m_2+1) \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ m_1+m_2 \\ m_1+m_2 \end{pmatrix}$$

We will solve the system using the Cramer formula. Denote by $\mathbf{w_i}$ the i-th column of the matrix Λ_2, $i = 1, 2, 3$, and set $\mathbf{w_R} = (0, m_1+m_2, m_1+m_2)^T$. Set also $A_1 = (\mathbf{w_R}, \mathbf{w_2}, \mathbf{w_3})$, $A_2 = (\mathbf{w_1}, \mathbf{w_R}, \mathbf{w_3})$ and $A_3 = (\mathbf{w_1}, \mathbf{w_2}, \mathbf{w_R})$ (these are (3×3) rational matrices). Then

$$x_0 = \frac{\det A_1}{\det \Lambda_2} = \frac{{m_1}^2 m_2 + 2\, m_1\, m_2 - 2\, m_1\, \lambda_{1,2} - m_1\, {m_2}^2 + 2\, m_2\, \lambda_{1,2}}{m_1\, m_2},$$

$$x_1 = \frac{\det A_2}{\det \Lambda_2} = \frac{m_2(m_1-m_2)(m_1+m_2)}{-\frac{1}{2}m_1m_2(m_1+m_2)} = -2\frac{m_1-m_2}{m_1} = \frac{2m_2}{m_1} - 2,$$

$$x_2 = \frac{\det A_3}{\det \Lambda_2} = \frac{m_1(m_2-m_1)(m_2+m_1)}{-\frac{1}{2}m_1m_2(m_2+m_1)} = -2\frac{m_2-m_1}{m_2} = \frac{2m_1}{m_2} - 2.$$

Further consider the following system (x_0, x_1, x_2, x_3 are the unknown variables and $P = m_1 + m_2 + m_3$)

$$\begin{pmatrix} 0 & m_1 & m_2 & m_3 \\ m_1 & \frac{1}{2}m_1(m_1+1) & \lambda_{1,2} & m_1m_3 - \lambda_{3,1} \\ m_2 & m_1m_2 - \lambda_{1,2} & \frac{1}{2}m_2(m_2+1) & \lambda_{2,3} \\ m_3 & \lambda_{3,1} & m_2m_3 - \lambda_{2,3} & \frac{1}{2}m_3(m_3+1) \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ P \\ P \\ P \end{pmatrix}$$

Using the Cramer formula, we obtain:

$$x_0 = P\frac{B}{A^2 + m_1m_2m_3P}$$

$$x_1 = -2P\frac{(m_3 - m_2)A + m_2m_3(2m_1 - m_2 - m_3)}{A^2 + m_1m_2m_3P}$$
$$x_2 = -2P\frac{(m_1 - m_3)A + m_1m_3(2m_2 - m_1 - m_3)}{A^2 + m_1m_2m_3P}$$
$$x_3 = -2P\frac{(m_2 - m_1)A + m_1m_2(2m_3 - m_1 - m_2)}{A^2 + m_1m_2m_3P}$$

where

$$A = 2\lambda_{1,2}m_3 + 2\lambda_{2,3}m_1 + 2\lambda_{3,1}m_2 - 3m_1m_2m_3,$$
$$B = -(1 + m_1m_2 + m_2m_3 + m_1m_3)A +$$
$$+2\lambda_{1,2}m_3(1 - 3m_1m_2 - m_1 + m_2 + 2\lambda_{1,2} + 2\lambda_{2,3} + 2\lambda_{3,1}) +$$
$$+2\lambda_{2,3}m_1(1 - 3m_2m_3 - m_2 + m_3 + 2\lambda_{1,2} + 2\lambda_{2,3} + 2\lambda_{3,1}) +$$
$$+2\lambda_{3,1}m_2(1 - 3m_1m_3 - m_3 + m_1 + 2\lambda_{1,2} + 2\lambda_{2,3} + 2\lambda_{3,1}).$$

As we will see in the next section, unique solutions $(x_0, \ldots, x_n)^T$ ($n = 2$ or $n = 3$) of previous systems of linear equations actually equal to a value $t_0(\mu_0)$ and to values of the mapping v from the table triple $(v, t_0, \mu_0)_N^n$, i.e. $x_0 = t_0(\mu_0)$ and $x_i = v(i)$ for all $i = 1, \ldots, n$.

5.2 Applications and examples

In this section we will apply results from the previous section to the theory of table triples and persistent states. We will also show some more examples of persistent table triples and prove that no reduced monotonous persistent 2-state exists. In the end, we will make a review of relations between persistent states, table triples and equivalences on an linearly ordered set.

Proposition 52. *Let $(v, t, \mu)_N^n$ be a persistent table triple with period P. Then*

$$\begin{pmatrix} 0 & m_1 & m_2 & \ldots & m_n \\ m_1 & \lambda_{1,1} & \lambda_{1,2} & \ldots & \lambda_{1,n} \\ m_2 & \lambda_{2,1} & \lambda_{2,2} & \ldots & \lambda_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_n & \lambda_{n,1} & \lambda_{n,2} & \ldots & \lambda_{n,n} \end{pmatrix} \cdot \begin{pmatrix} t_0(\mu_0) \\ v(1) \\ v(2) \\ \vdots \\ v(n) \end{pmatrix} = \begin{pmatrix} 0 \\ P \\ P \\ \vdots \\ P \end{pmatrix}$$

Proof: The proposition follows directly from Corollary 40 and Lemma 44. □

Without any results about equivalences we can prove the following lemma.

Lemma 53. *The mapping μ of a persistent table triple $(v, t_0, \mu_0)_N^n$ determines the mapping t_0 and the value μ_0.*

Proof: Let $\sim$ be the equivalence determined by μ, and set n as the number of equivalence classes of $\sim$. Let us imagine an empty table with n columns and $P+1$ rows (indexed from zero) and an empty header. Denote equivalence classes of $\sim$ by $C_1, \ldots, C_n$. For each $j \in C_i$ mark the position in the j-th row and the i-th column. In the zero row mark the position i such that $P \in C_i$. Set the value 1 above each marked position (if this appears in the table). Besides the header it is now easy to fill all other positions in the table, since we know that in every column all sequences between the marked positions are strictly decreasing by one, whenever the table is determined by a persistent table triple.

The zero row of the constructed table defines the mapping t_0, and $\mu_0 = i$ such that $P \in C_i$.

□

Theorem 54. *The mapping μ of a persistent table triple completely determines the table triple.*

Proof: We have already shown that given a mapping μ of a persistent table triple $T_0 = (v, t_0, \mu_0)_N^n$ we can uniquely determine the equivalence $\sim$ on the set Δ_P and consequently the matrix Λ_n. Proposition 52 and Theorem 49 imply that the values $v(i),\ i = 1, \ldots n$ can be uniquely determined from the matrix Λ_n and Lemma 53 impliest that μ uniquely determines the mapping t_0 and the value μ_0. □

Proposition 55. *Let $N \in \mathbb{N}$ or $n = \infty$. Then no reduced monotonous persistent 2-state (i, j, σ) exists.*

Proof: Proposition 26 implies that each such a state would uniquely determine a persistent table triple with an injective mapping v. Assume we have such a table triple, e.g. $(v, t_0, \mu_0)_N^n$. Then Proposition 52 implies

$$\begin{pmatrix} 0 & m_1 & m_2 \\ m_1 & \frac{1}{2}m_1(m_1+1) & \lambda_{1,2} \\ m_2 & m_1m_2 - \lambda_{1,2} & \frac{1}{2}m_2(m_2+1) \end{pmatrix} \cdot \begin{pmatrix} t_0(\mu_0) \\ v(1) \\ v(2) \end{pmatrix} = \begin{pmatrix} 0 \\ m_1+m_2 \\ m_1+m_2 \end{pmatrix}$$

At the end of previous section we have shown, that the unique solution of this system gives us

$$v(1) = \frac{2m_2}{m_1} - 2, \quad v(2) = \frac{2m_1}{m_2} - 2,$$

Since $v : \Delta_n \to \mathbb{Z}\setminus\{-1\}$, we see that m_1 has to divide $2m_2$ and also m_2 has to divide $2m_1$. So we get $m_1x = 2m_2$ and $m_1y = 2m_1$ for some $x, y \in \mathbb{N}$. Thus $m_1xy = 4m_1$ and $xy = 4$. Either $x = y = 2$ or $\{x, y\} = \{1, 4\}$. If $x = 1, y = 4$

then $2m_2 = m_1$ and $v(1) = \frac{2m_2}{m_1} - 2 = -1$, if $x = 4, y = 1$ then $v(2) = -1$. Both these results are in contradiction with $v(\Delta_n) \subseteq \mathbb{Z} \setminus \{-1\}$. So $x = y = 2$. Thus it is $v(1) = v(2)$ and this is in contradiction with the injectivity of v. □

Conjecture 56. *Assume $N = \infty$. No persistent n-state (i, j, σ) exists for any $n > 1$.*

Remark 57. *If this conjecture does not hold, i.e. if there exists some persistent n-state (i_0, j_0, σ_0), it could be easily transferred to the situation $N \in \mathbb{N}$, $N \geq max\{\eta_k(a) \mid a \in \Sigma, 0 \leq k \leq P\}$.*

Later on (Section 5.3) we will prove this conjecture for cases $n = 3$ and $n = 4$.

Example 58. Let $N \geq 6$. Figure 10 shows a persistent table triple $(v, t_0, \mu_0)_N^4$ with $v = (-2, 1, 1, 1)$, $t_0 = (1, 2, 3, 5)$, $\mu_0 = 4$. This table triple has period equal to 9.

-2	1	1	1
1	2	3	**5**
3	1	2	4
2	**4**	1	3
1	3	**5**	2
3	2	4	1
2	1	3	**4**
1	**5**	2	3
3	4	1	2
2	3	**4**	1
1	2	3	**5**

Figure 10: A persistent table with non-injective mapping v (Example 58).

Example 59. Let $N \geq 7$. Figure 11 shows a persistent table triple $(v, t_0, \mu_0)_N^4$ with $v = (-2, -2, 4, 4)$, $t_0 = (1, 2, 3, 6)$, $\mu_0 = 4$. This table triple has period equal to 6.

Example 60. Let $N \geq 7$. Figure 12 shows a persistent table triple $(v, t_0, \mu_0)_N^5$ with $v = (-3, 3, 0, 3, 0)$, $t_0 = (1, 2, 3, 5, 6)$, $\mu_0 = 5$. This table triple has period equal to 6.

Let $n \in \mathbb{N}$. In Proposition 26 we have shown that each reduced monotonous persistent n-state (i, j, σ) uniquely determines a persistent table triple $(v, t, \mu)_N^n$. Furthermore, each persistent table triple with period P uniquely determines

-2	-2	4	4
1	2	3	**6**
4	1	2	5
3	**2**	1	4
2	1	**6**	3
1	**4**	5	2
2	3	4	1
1	2	3	**6**

Figure 11: A persistent table with non-injective mapping v (Example 59).

-3	3	0	3	0
1	2	3	5	**6**
3	1	2	4	5
2	**6**	1	3	4
1	5	**6**	2	3
3	4	5	1	2
2	3	4	**6**	1
1	2	3	5	**6**

Figure 12: A persistent table with non-injective mapping v (Example 60).

an equivalence $\sim$ on Δ_P with n equivalence classes. Let φ denote the mapping from the set of all persistent table triples $(v,t,\mu)_N^n$ (denote this set $\mathcal{T}_n$) to the set $\mathcal{E}_n = \{(P,\sim) \mid P \in \mathbb{N},\ \sim \textit{ an equivalence on } \Delta_P \textit{ with } n \textit{ equivalence classes}\}$,

$$\varphi : \mathcal{T}_n \to \mathcal{E}_n.$$

It is easy to see that the mapping φ is injective (later on we will describe an inverse mapping $\varphi^{-1} : \varphi(\mathcal{T}_n) \to \mathcal{T}_n$, i.e. we will show how to construct a table triple out of any $(P,\sim) \in \varphi(\mathcal{T}_n)$).

In Section 5.1 we have shown that each couple $(P,\sim) \in \mathcal{E}_n$ uniquely determines a matrix $\Lambda_n \in (\mathbb{R})_{(n+1)\times(n+1)}$. We will denote this mapping by χ,

$$\chi : \mathcal{E}_n \to (\mathbb{R})_{(n+1)\times(n+1)}.$$

Later on (Example 61) we will show that χ is not necessarily injective. However it might be injective on the set $\varphi(\mathcal{T}_n)$. Set $\mathcal{M}_n = \chi(\mathcal{E}_n)$.

Theorem 49 states that the matrix Λ_n is regular for each $(P,\sim) \in \mathcal{E}_n$. Thus the matrix Λ_n uniquely determines a vector $(s, v(1), \ldots, v(n)) \in \mathbb{R}^{n+1}$. So we obtain a mapping ω from the set $\mathcal{M}_n$ to $\mathbb{R}^{n+1}$,

$$\omega : \mathcal{M}_n \to \mathbb{R}^{n+1}.$$

Assume $(v,t_0,\mu_0)_N^n$ is a persistent table triple with period P. Denote the unique solution of the system of linear equations determined by the matrix

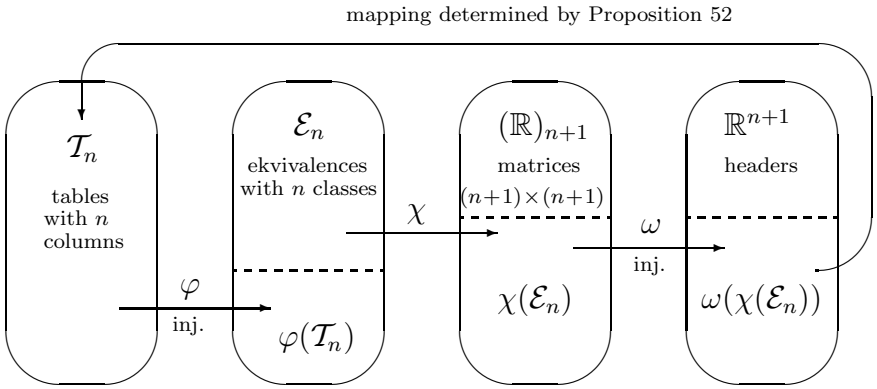


Figure 13: A graphical illustration of mappings φ, χ and ω

$\Lambda = \chi(\varphi((v, t_0, \mu_0)_N^n))$ by $(r, w_1, w_2, \ldots, w_n)$, i.e.

$$(r, w_1, w_2, \ldots, w_n)^T = \omega(\chi(\varphi((v, t_0, \mu_0)_N^n))).$$

Then Proposition 52 implies

$$(r, w_1, w_2, \ldots, w_n) = (t_0(\mu_0), v(1), v(2), \ldots, v(n)).$$

Example 61. Let $n = 2$ and let $\sim_1$ be an equivalence on Δ_5 with equivalence classes $C_1 = \{1, 3, 5\}$, $C_2 = \{2, 4\}$ and let $\sim_2$ be another equivalence on Δ_5 with equivalence classes $D_1 = \{2, 3, 4\}$, $D_2 = \{1, 5\}$. Each of these equivalences determines the matrix

$$\Lambda_2 = \begin{pmatrix} 0 & 3 & 2 \\ 3 & 3 & 6 \\ 2 & 3 & 3 \end{pmatrix}.$$

Example 62. Let $n = 4$ and $\sim$ be an equivalence on the set Δ_{15}. Let $C_1 = \{1, 5, 10, 14\}$, $C_2 = \{3, 7, 11, 13\}$, $C_3 = \{2, 4, 8, 12\}$, $C_4 = \{6, 9, 15\}$ be a decomposition of the set Δ_{15} to equivalence classes of $\sim$. Then

$$\Lambda_4 = \begin{pmatrix} 0 & 4 & 4 & 4 & 3 \\ 4 & 10 & 7 & 9 & 4 \\ 4 & 9 & 10 & 10 & 5 \\ 4 & 7 & 6 & 10 & 3 \\ 3 & 7 & 6 & 9 & 6 \end{pmatrix}.$$

Using $(0, 15, 15, 15)^T$ as the right-hand side vector we obtain a solution

$$(s, v(1), v(2), v(3), v(4))^T = (-\frac{475}{82}, \frac{60}{41}, -\frac{25}{82}, \frac{265}{82}, -\frac{35}{41})^T.$$

Clearly $\sim \notin \varphi(\mathcal{T}_4)$ since $v : S \to \mathbb{Z} \setminus \{-1\}$ does not hold.

Experimental results has shown that for a randomly chosen equivalence $\sim$, it is quite rare to obtain an integral solution. We will see (Example 64) that this condition is not sufficient for persistence. But firstly we will show how to uniquely determine a table triple out of any $(P, \sim) \in \varphi(\mathcal{T}_n)$ (part of this process was already described in the proof of Theorem 54).

Let $(P, \sim) \in \varphi(\mathcal{T}_n)$. Set n as the number of equivalence classes of $\sim$. So we can already draw an empty table with n columns and $P+1$ rows (indexed from zero) and an empty header. Denote equivalence classes of $\sim$ by $C_1, \ldots, C_n$. For each $j \in C_i$ mark the position in the j-th row and the i-th column. In the zero row mark the position i if $P \in A_i$. Set the value 1 above each marked position (if this appears in the table). Since in each table determined by a persistent table triple values in each column form decreasing sequences between marked positions, it is easy to fill the rest of the table besides the header of the table. The header can be uniquely determined out of the equations in Definition 19. The header of the constructed table defines the mapping v, first row defines the mapping t_0 and $\mu_0 = i$ if $P \in A_i$. So we have obtained a table triple $(v, t_0, \mu_0)^n_N$ which uniquely determines given equivalence and the value P. Example 63 illustrates this process.

Example 63. Let $\sim$ be an equivalence on a set Δ_4 with equivalence classes $C_1 = \{1, 3\},\ C_2 = \{2\},\ C_3 = \{4\}$. Figure 14 shows the construction of a table out of this equivalence.

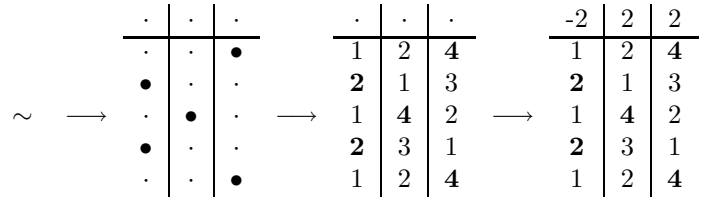


Figure 14: Construction of a table from an equivalence

Example 64. Let $\sim$ be an equivalence on a set Δ_{10} with equivalence classes $C_1 = \{2, 6\},\ C_2 = \{3\},\ C_3 = \{1, 5, 8, 10\},\ C_4 = \{4, 7, 9\}$. This equivalence

determines the matrix

$$\Lambda = \begin{pmatrix} 0 & 2 & 1 & 4 & 3 \\ 2 & 3 & 1 & 3 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 4 & 5 & 3 & 10 & 6 \\ 3 & 5 & 3 & 6 & 6 \end{pmatrix}$$

Using $(0, 10, 10, 10, 10)^T$ as the right hand side vector we obtain the solution

$$\mathbf{v} = (s, v(1), v(2), v(3), v(4))^T = (0, 2, 8, 0, -4)^T.$$

Figure 15 shows a table constructed by the technique described above. Since the obtained vector v is not compatible with this table, equivalence $\sim$ does not determine any persistent table triple (although this vector has integer coordinates and $v(i) \neq -1$ for all $1 \leq i \leq 4$). I.e.

$$\sim \notin \varphi(\mathcal{T}_n)$$

.

.	.	.	.
1	2	**4**	3
4	1	3	2
3	**9**	2	1
2	8	1	**3**
1	7	**3**	2
5	6	2	1
4	5	1	**2**
3	4	**2**	1
2	3	1	**4**
1	2	**4**	3

Figure 15: A table determined by equivalence $\sim$ from Example 64

5.3 Nonexistence of small states

In this section we will always assume $N = \infty$, if not stated otherwise.

The following lemma states that in the tabular model there cannot be two subsequent ones in the same column for any persistent state except the Finney state.

Lemma 65. *Let $T_0 = (v, t_0, \mu_0)_N^n$ be a persistent table triple with $n \geq 2$ and period P. Then for all $l \in \Delta_P$ we have $t_l(\mu_l) > 1$.*

Proof: Let us assume the opposite and let l be the smallest integer for which $t_l(\mu_l) = 1$ and $\mu_l = a \in \Delta_n$. Since $\mu_l = a$ if and only if $t_{l-1}(a) = 1$ we also have $t_{l-1}(\mu_l) = 1$. Consequently $t_{l-1}(b) > 1$ for all $b \in \Delta_n,\ b \neq a$.

Definition 19 implies that $t_l(a) = t_{l-1}(a) + v_a = 1$. Since $t_{l-1}(\mu_{l-1}) \geq 2$ we get $v_a < 0$. Now, $\mu_{l+1} = a$ $(t_l(a) = 1)$ and $t_{l+1}(a) = t_l(\mu_l) + v_a = t_l(a) + v_a = 1 + v_a < 1$ is a contradiction. □

Proposition 66. *Let $T_0 = (v, t_0, \mu_0)_N^n$ be a persistent table triple with $n \geq 2$ and period P. Then for all $l \in \Delta_P$ there exists $a \in \Delta_n$ for which $t_l(a) = 2$.*

Proof: Assume $2 \notin \mathrm{Im}(t_l)$ for some $l \in \Delta_P$. Thus for $b \in \Delta_n$ either $t_l(b) = 1$ or $t_l(b) > 2$. Lemma 65 implies that $\mu_l \neq 1$, and so $t_l(\mu_l) > 2$. Denote by a the element of Δ_n for which $t_l(a) = 1$. We have $t_{l+1}(b) \geq 2$ for all $b \neq a$ and since T_0 is persistent one gets $t_{l+1}(a) = t_l(\mu_l) + v_a = 1$. At the same time $\mu_{l+1} = a$ since $t_l(a) = 1$. This contradicts Lemma 65. □

Lemma 67. *Let $T_0 = (v, t_0, \mu_0)_N^n$ be a persistent table triple with period P. Set $\delta = max\{t_i(a) \mid 0 \leq i \leq P,\ a \in \Delta_n\}$. If $t_l(a) = \delta$, where $0 \leq l \leq P$, then $\mu_l = a$.*

Proof: The existence of δ follows directly from the periodicity of the table triple. Assume $t_l(a) = \delta$ and $a \neq \mu_l$. Then $t_{l-1}(a) = \delta + 1$ which contradicts the choice of δ. □

Proposition 68. *There exists no persistent table triple $(v, t_0, \mu_0)_\infty^3$ with an injective mapping v.*

Proof: Let us assume that $T_0 = (v, t_0, \mu_0)_\infty^3$ is a persistent table triple. Proposition 30 implies that T_0 is periodic (with period P). Set $\delta = max\{t_i(a) \mid 0 \leq i \leq P,\ a \in \Delta_n\}$. Firstly we will prove the proposition for $\delta = 3$ and $\delta = 4$ and then for all $\delta \geq 5$.

a) Assume $\delta = 3$. Order the columns so that $t_0 = (1, 2, 3)$. Then $\mu_0 = 3$, by Lemma 67. Then $t_1 = (3 + v_1, 1, 2)$ and $\mu_1 = 1$. Since t_1 has to be injective (Lemma 22) and $\delta = 3$ we get $v_1 = 0$. Thus $t_2 = (2, 3 + v_2, 1)$ and we obtain that $v_2 = 0 = v_1$. The mapping t_2 is not injective and T_0 is not persistent.

b) Assume $\delta = 4$. Lemma 67 and Proposition 66 implies that $t_0 = (1, 2, 4)$ and that $\mu_0 = 3$. Then $\mu_1 = 1$ and $t_1 = (v_1 + 4, 1, 3)$. Proposition 66 implies $v_1 + 4 = 2$ and so $v_1 = -2$. Then $\mu_2 = 2$ and $t_2 = (1, 2 + v_2, 2)$ thus we have $v_2 \in \{1, 2\}$. Since $t_3 = (v_2, 1 + v_2, 1)$ necessarily $v_2 = 2$. Then $t_4 = (1, 2, 2 + v_3)$. Since $\delta = 4$ and $v_2 = 2$ we have $v_3 = 1$ (if we set $v_2 = 2$ we obtain the table triple from Example 32). Consequently $t_5 = (1, 1, 2)$ and Lemma 22 implies T_0 is not persistent.

c) Assume $\delta \geq 5$. Order the columns so that $t_0 = (1, 2, \delta)$ and $\mu_0 = 3$. Then $\mu_1 = 1$ and $t_1 = (\delta + v_1, 1, \delta - 1)$. Since $\delta \geq 5$ Proposition 66 implies

$\delta + v_1 = 2$ and so $v_1 = -(\delta - 1) \leq -3$. Then $\mu_2 = 2,\ t_2 = (1, 2 + v_2, \delta - 2)$ and since $\delta - 2 \geq 3$ we have $2 + v_2 = 2$ so $v_2 = 0$. Finally $\mu_3 = 1$ and $t_3(1) = t_2(\mu_2) + v_1 = 2 - (\delta - 2) = 4 - \delta \leq -1$. Hence T_0 is not persistent (Definition 19 implies $t_i(a) \geq 1$ for all i and all a). □

Corollary 69. *Assume $N = \infty$. Then there exists no reduced persistent 3-state (i, j, σ).*

Proposition 70. *There exists no persistent table triple $(v, t_0, \mu_0)^4_\infty$ with an injective mapping v.*

Proof: Assume the opposite and let $T_0 = (v, t_0, \mu_0)^4_\infty$ be a persistent table triple. Set $\delta = max\{t_i(a) \mid 0 \leq i \leq P,\ a \in \Delta_n\}$. We will consider sequentially cases $\delta = 4,\ \delta = 5,\ \delta = 6$ and $\delta \geq 7$.

I) $\delta = 4$: Assume $T_0 = (v, t_0, \mu_0)^4_\infty$ is a persistent table triple with $t_0 = (1, 2, 3, 4)$. Then $\mu_0 = 4$, by Lemma 67. We have $\mu_1 = 1$ and $t_1 = (4 + v_1, 1, 2, 3)$, and the maximality of δ implies $v_1 = 0$. Then $\mu_2 = 2$ and $t_2 = (3, 4 + v_2, 1, 2)$, so necessarily $v_2 = 0$. This is in contradiction with the injectivity of t_2.

II) $\delta = 5$: We will distinguish two different situations.

II.1) Let $t_0 = (1, 2, 3, 5)$. Then $t_1 = (5 + v_1, 1, 2, 4),\ \mu_1 = 1$ and therefore $v_1 = 0$ or $v_1 = -2$.

II.1.1) Assume $v_1 = 0$. Then $t_2 = (4, 5+v_2, 1, 3),\ \mu_2 = 2$ and Proposition 66 implies $v_2 = -3$. So $t_3 = (3, 1, 2 + v_3, 2)$ and $\mu_3 = 3$. We see that either $v_3 = 2$ or $v_3 = 3$. In the former case we obtain $t_4(2) = t_3(\mu_3) + v_2 = 4 - 3 = 1$, which contradicts $\mu_4 = 2$, by Lemma 65. In the latter case $t_4(2) = t_4(1) = 2$, again a contradiction.

II.1.2) Assume $v_1 = -2$. Then $t_2 = (2, 3 + v_2, 1, 3)$ and so either $v_2 = 1$ or $v_2 = 2$.

II.1.2.1) Assume $v_2 = 1$. We get $t_3 = (1, 3, 4 + v_3, 2), \mu_3 = 3$ and so $v_3 = 0$ since $v_2 = 1$. Then $t_4 = (2, 2, 3, 1)$ and so t_4 in not injective.

II.1.2.2) Consider the latter case, $v_2 = 2$. Then $t_3 = (1, 4, 5 + v_3, 2)$ and $\mu_3 = 3$. Obviously either $v_3 = 0$ or $v_3 = -2$. Since already $v_1 = -2$ we have $v_3 = 0$. Then $t_4 = (3, 3, 4, 1)$ and T_0 is not persistent.

II.2) Let $t_0 = (1, 2, 4, 5)$. Then $t_1 = (5 + v_1, 1, 3, 4),\ \mu_1 = 1$ and Proposition 66 implies $v_1 = -3$. So $\mu_2 = 2,\ t_2 = (1, 2 + v_2, 2, 3)$ and hence either $v_2 = 2$ or $v_2 = 3$. If $v_2 = 2$ then $t_2(2) = 4$ and $t_3 = (1, 3, 1, 2)$. If $v_2 = 3$ then $t_2(2) = 5$ and $t_3 = (2, 3, 1, 2)$. So there exists no $T_0 = (v, t_0, \mu_0)^4_\infty$ with an injective mapping v and $\delta = 5$.

III) $\delta = 6$: There are three possibilities for the mapping t_0. We will consider them sequentially.

III.1) Let $t_0 = (1, 2, 3, 6)$. Then $t_1 = (6 + v_1, 1, 2, 5)$, $\mu_1 = 1$ and so $v_1 \in \{-3, -2, 0\}$.

III.1.1) Assume $v_1 = -3$. So $t_1 = (3, 1, 2, 5)$ and $\mu_1 = 1$. Then $t_2 = (2, 3 + v_2, 1, 4)$, $\mu_2 = 2$ and thus $v_2 \in \{0, 2, 3\}$.

III.1.1.1) Assume $v_2 = 0$. Then $t_2 = (2, 3, 1, 4)$, $\mu_2 = 2$ and $t_3 = (1, 2, 3 + v_3, 3)$, $\mu_3 = 3$. So $v_3 \in \{1, 2, 3\}$. If $v_3 = 1$ then $t_4(1) = 1$ and because $\mu_4 = 1$ Lemma 65 implies T_0 is not persistent. If $v_3 = 2$ then $t_4 = (2, 1, 4, 2)$ and Lemma 22 implies that T_0 is not persistent. Hence $v_3 = 3$ and $t_4 = (3, 1, 5, 2)$, $\mu_4 = 1$. Then $t_5 = (2, 3, 4, 1)$, $\mu_5 = 2$ and $t_6 = (1, 2, 3, 3 + v_4)$, $\mu_6 = 4$. So either $v_4 = 1$ or $v_4 = 2$. In the former case ($v_4 = 1$) $t_7(1) = t_6(\mu_6) + v_1 = 4 - 3 = 1$ and since $\mu_7 = 1$ this is in contradiction with Lemma 65. Assume $v_4 = 2$. Then $t_7 = (2, 1, 2, 4)$ and T_0 is not persistent. Notice, that if we set $v_4 = 3$ we obtain the persistent table triple described in Example 33.

III.1.1.2) Assume $v_2 = 2$. Then $t_2 = (2, 5, 1, 4)$, $\mu_2 = 2$ and $t_3 = (1, 4, 5 + v_3, 3)$, $\mu_3 = 3$. Proposition 66 implies $5 + v_3 = 2$ and hence $v_3 = -3 = v_1$ hence v is not injective.

III.1.1.3) Assume $v_2 = 3$. Then $t_2 = (2, 6, 1, 4)$, $\mu_2 = 2$ and $t_3 = (1, 5, 6 + v_3, 3)$, $\mu_3 = 3$. Proposition 66 implies $6 + v_3 = 2$ and hence $v_3 = -4$. Then $t_4(1) = t_3(\mu_3) + v_1 = 2 - 3 = -1$ which is in a contradiction with $t_l(a) \geq 1$, which we assumed in the case $N = \infty$.

III.1.2) Assume $v_1 = -2$. Then $t_1 = (4, 1, 2, 5)$, $\mu_1 = 1$ and $t_2 = (3, 4 + v_2, 1, 4)$. Proposition 66 implies $4 + v_2 = 2$ and hence $v_2 = -2 = v_1$, so v is not injective.

III.1.3) Assume $v_1 = 0$. Then $t_1 = (6, 1, 2, 5)$, $\mu_1 = 1$ and $t_2 = (5, 6 + v_2, 1, 4)$, $\mu_2 = 2$. Proposition 66 implies $6 + v_2 = 2$ and hence $v_2 = -4$. We have $t_3 = (4, 1, 2 + v_3, 3)$ and necessarily $v_3 = 0 = v_1$.

III.2) Let $t_0 = (1, 2, 4, 6)$. Hence $t_1 = (6 + v_1, 1, 3, 5)$, $\mu_1 = 1$ and so $v_1 + 6 = 2$, $v_1 = -4$. Then $t_2 = (1, 2 + v_2, 2, 4)$, $\mu_2 = 2$ and hence $v_2 \in \{1, 3, 4\}$. If $v_2 = 1$ then $t_3(1) = t_2(\mu_2) + v_1 = 3 - 4 = -1$ and T_0 is not persistent. If $v_2 = 3$ then $t_3(1) = t_2(\mu_2) + v_1 = 5 - 4 = 1$ and because $\mu_3 = 1$ this is in contradiction with Lemma 65. Finally assume $v_2 = 4$. Then $t_3 = (2, 5, 1, 3)$, $\mu_3 = 1$ and $t_4 = (1, 4, 2 + v_3, 2)$, $\mu_4 = 3$. Therefore $v_3 \in \{1, 3, 4\}$, but since $v_2 = 4$ either $v_3 = 1$ or $v_3 = 3$. If $v_3 = 1$ then $t_5(1) = t_4(\mu_4) + v_1 = 3 - 4 = -1$. If $v_3 = 3$ then $t_5(1) = t_4(\mu_4) + v_1 = 5 - 4 = 1$ and Lemma 65 implies that T_0 in not persistent.

III.3) Let $t_0 = (1, 2, 5, 6)$. Hence $t_1 = (6 + v_1, 1, 4, 5)$, $\mu_1 = 1$ and Proposition 66 implies $v_1 = -4$. Then $t_2 = (1, 2 + v_2, 3, 4)$, $\mu_2 = 2$ and so $v_2 = 0$. Then $t_3(1) = t_2(\mu_2) + v_1 = 2 - 4 = -2$ and T_0 is not persistent. We have proved that there exists no $T_0 = (v, t_0, \mu_0)^4_\infty$ with an injective mapping v and $\delta = 6$.

IV) $\delta \geq 7$:

IV.1) Assume $t_0 = (1, 2, 3, \delta)$, $\mu_0 = 4$. Then $t_1 = (\delta + v_1, 1, 2, \delta - 1)$ and $\mu_1 = 1$. We will now consider all possibilities for the value $\delta + v_1$.

IV.1.1) Assume $\delta + v_1 = 3$, so $v_1 = -(\delta - 3) \leq -4$. Therefore $t_2 = (2, 3 + v_2, 1, \delta - 2)$, $\mu_2 = 2$ and $v_2 \neq -1$, $v_2 \neq -2$. Then $t_3 = (1, 2 + v_2, 3 + v_2 + v_3, \delta - 3)$, $\mu_3 = 3$ and Proposition 66 implies that either $2 + v_2 = 2$ or $3 + v_2 + v_3 = 2$. In the latter case we get $t_4(1) = t_3(\mu_3) + v_1 = 2 + v_1 \leq -2$ and T_0 is not persistent. Thus it is $v_2 = 0$ and $t_3 = (1, 2, 3 + v_3, \delta - 3)$. Therefore $t_4 = (3 + v_1 + v_3, 1, 2 + v_3, \delta - 4)$. We have $t_4(1) = 3 + v_1 + v_3 = 3 - (\delta - 3) + v_3 = 6 - \delta + v_3 \leq v_3 - 1$. If $v_3 \leq 2$ then $t_4(1) \leq v_3 - 1 \leq 1$ and since $\mu_4 = 1$ this contradicts Lemma 65. So it is $v_3 > 2$ and hence $2 + v_3 \geq 5$. So necessarily $3 + v_1 + v_3 = 2$ and so $v_3 = -1 - v_1 = \delta - 4$. Hence $t_4 = (2, 1, \delta - 2, \delta - 4)$, $\mu_4 = 1$ and $t_5 = (1, 2, \delta - 1, \delta - 5)$, $\mu_5 = 2$. Then $t_6(1) = t_5(\mu_5) + v_1 = 2 - (\delta - 3) < 0$.

IV.1.2) Assume $\delta + v_1 = 4$. Then $t_2 = (3, 4 + v_2, 1, \delta - 2)$, $\mu_2 = 2$ and since $\delta - 2 \geq 5$ Proposition 66 implies $v_2 = -2$. Consequently $t_3 = (2, 1, 2 + v_3, \delta - 3)$, $\mu_3 = 3$ and $t_4 = (1, v_3, v_3 + 1, \delta - 4)$, $\mu_4 = 2$. Since $\delta \geq 7$ necessarily $v_3 = 2$. Therefore $t_5(1) = t_4(\mu_4) + v_1 = 2 - (\delta - 4) = 6 - \delta < 0$ and T_0 is not persistent.

IV.1.3) Assume $\delta + v_1 \geq 5$ so $v_1 \geq -(\delta - 5)$. Then $t_1 = (\delta + v_1, 1, 2, \delta - 1)$, $\mu_1 = 1$ and $t_2 = (\delta + v_1 - 1, \delta + v_1 + v_2, 1, \delta - 2)$, $\mu_2 = 2$. Since $\delta \geq 7$ Proposition 66 implies $\delta + v_1 + v_2 = 2$ and then $2 = \delta + v_1 + v_2 \geq 5 + v_2$ implies $v_2 \leq -3$. We have $t_3 = (\delta + v_1 - 2, 1, 2 + v_3, \delta - 3)$ and since $\delta + v_1 - 2 \geq 3$ it is necessarily $v_3 = 0$. Finally $t_4(2) = t_3(\mu_3) + v_2 = 2 + v_2 \leq -1$ and this is in contradiction with the persistent property of T_0.

IV.2) Assume $t_0 = (1, 2, 4, \delta)$, $\mu_0 = 4$. Then $t_1 = (\delta + v_1, 1, 3, \delta - 1)$ and $\mu_1 = 1$. Proposition 66 implies $\delta + v_1 = 2$ and hence $v_1 = -(\delta - 2)$ and $t_2 = (1, 2 + v_2, 2, \delta - 2)$, $\mu_2 = 2$. Then $t_3 = (2 + v_1 + v_2, 1 + v_2, 1, \delta - 3)$, $\mu_3 = 1$ and either $2 + v_1 + v_2 = 2$ or $1 + v_2 = 2$. In the latter case we obtain $v_2 = 1$ and hence $2 + v_1 + v_2 = 3 + v_1 = 3 - (\delta - 2) = 5 - \delta < 0$. Therefore necessarily $2 + v_1 + v_2 = 2$ and so $v_2 = \delta - 2$. Then $t_4 = (1, \delta - 2, 2 + v_3, \delta - 4)$, $\mu_4 = 3$ and Proposition 66 implies $2 + v_3 = 2$. We obtain $t_5(1) = t_4(\mu_4) + v_1 = 2 - (\delta - 2) = 4 - \delta < 0$.

IV.2) Assume $t_0 = (1, 2, a, \delta)$, $\mu_0 = 4$ for some $5 \leq a < \delta$. Then $t_1 = (\delta + v_1, 1, a - 1, \delta - 1)$, $\mu_1 = 1$ and Proposition 66 implies $\delta + v_1 = 2$ so it is $v_1 = -(\delta - 2)$. Consequently $t_2 = (1, 2 + v_2, a - 2, \delta - 2)$, $\mu_2 = 2$ and because $a \geq 5$ it is $2 + v_2 = 2$. Finally $t_3(1) = t_2(\mu_2) + v_1 = 2 + v_1 = 4 - \delta < 0$. □

Corollary 71. *Assume $N = \infty$. Then no reduced persistent 4-state (i, j, σ) exists.*

5.4 Cryptanalytical significance

In this section we will focus on cryptanalytical applications of persistent states. Firstly we will mention an attack based on the Finney state, then indicate a generalised attack based on any persistent state and finally we will show an example of such an attack.

In 1994, Hal Finney described the so called Finney state in [Fi94] (see Example 27), which is the only known persistent state. We will now describe another interesting property of this state (following [Ma01]).

Assume we have a 2-state (i_0, j_0, σ_0) $(3 \leq N < \infty)$, where $i_0 = 0,\ j_0 = 1$ and $\sigma_0(1) = 1, \sigma_0(2) = a$. Clearly, (i_0, j_0, σ_0) is an extension of the Finney state (which is an 1-state). Then $i_1 = 1,\ j_1 = 2$ and $\sigma_1(2) = 1, \sigma_1(1) = a$. For all $1 < k < N$ we have $\sigma_k(1) = a$, because $i_k \neq 1 \neq j_k$ for all these k. Consequently $i_N = 0,\ j_N = 1$ and $\sigma_N(1) = 1,\ \sigma_N(0) = a$.

Let an n-state (i_0, j_0, σ_0) be an extension of the Finney state with $n = N,\ i_0 = 0,\ j_0 = 1,\ \sigma_0(1) = 1$. From the above it follows

$$\forall 0 \leq i < N\ \forall k \geq 0 \text{ if } \sigma_k(i) \neq 1 \text{ then } \sigma_{k+N}(i - 1 \bmod N) = \sigma_k(i). \tag{14}$$

Since σ_0 is a permutation of the set $\{0, \ldots, N-1\}$, there exists a value $l \neq 1$ such that $\sigma_0(l) = N - 1$. Assume $\sigma_0(0) = a \neq N - 1$ and $\sigma_0(2) = b \neq N - 1$. Denote an output word of the PRGA in round i as z_i. We get $\sigma_{l-1}(i_{l-1}) = \sigma_{l-1}(l-1) = N - 1$ in the round $N - 1$ and the output of the PRGA equals

$$z_{l-1} = \sigma_{l-1}(\sigma_{l-1}(i_{l-1}) + \sigma_{l-1}(j_{l-1}) \bmod N) = \sigma_{l-1}(0) = a.$$

It follows from Equation 14, that (recall that $\sigma_0(1) = 1$) $\sigma_{l-1+N-1}(l-2) = N-1$ and

$$z_{l-1+N-1} = \sigma_{l-1+N-1}(i_{l-1+N-1} + j_{l-1+N-1} \bmod N) = \sigma_{l-1+N-1}(0) = b$$

. Generally for all $1 \leq k \leq N - 1$ we get

$$\sigma_{l-1+k(N-1)}(l - 1 - k \bmod N) = N - 1,$$

$$z_{l-1+k(N-1)} = \sigma_{l-1+k(N-1)}(0) = \sigma_0(k) \tag{15}$$

Thus we can determine the initial permutation σ_0 just by observing the output sequence.

The PRGA of RC4 sets $i_0 = j_0 = 0$ so the Finney state can never occur in the PRGA, i.e. this state can never appear in real RC4 streams (authors of [BGN05] described a fault analysis[4] of RC4 using the Finney state).

[4]Fault analysis attacks are attacks based on deliberately introducing faults into cryptographic processors in order to determine the secret keys or the inner state of a cipher.

Let $T_0 = (i_0, j_0, \sigma_0)$ be a persistent n-state with period P. Assume we have a full inner state of RC4 (i, j, S) $(i, j \in \mathbb{Z}_N,\ S \in \mathcal{S}_N)$ where $i = i_0,\ j = j_0$, and σ_0 is a restriction of S. Since T_0 is persistent, all following RC4 states given by PRGA are independent of the value $S(a)$ for all $a \in \{0, \ldots, N-1\} \setminus \mathrm{Dom}(\sigma_0)$. As in the case of the Finney state (which has period equal to one), the state T_0 behaves identically on all $a,\ a \notin \mathrm{Dom}(\sigma_0)$ each P steps, i.e. it changes the permutation S in a predictable way. Consequently this predictable changes can be used to obtain a part of the inner state of RC4. Therefore, discovery of a persistent state would be an important step in the cryptanalysis of RC4.

Example 72 shows how the persistent state from Example 32 can be used to determine the secret inner permutation of RC4.

Example 72. *Let us have $N = 256$ (in the rest of this example all additions are carried out mod N). Assume we have RC4 in the state (i_0, j_0, S_0) (subscript 0 does not mean this is an initial state with $i_0 = 0 = j_0$), where*

$$i_0 = l,\ j_0 = l + 4,\ S_0(l + 1) = 1,\ S_0(l + 2) = 3,\ S_0(l + 4) = 3$$

S_0 is obviously not a permutation, but assume we have such an RC4 state (e.g. this state can be reached by a fault injection). In Example 32 we have seen that a table triple given by this 3-state is persistent and Proposition 26 implies that this state is persistent. For simplicity assume $0 \leq l < N - 10$. Let us have

$$S_0 : (l + 5, l + 6, l + 7, l + 8, l + 9, l + 10) \longmapsto (a, b, c, d, e, f).$$

Denote the inner permutation of RC4 in the round k by S_k. Analysis of this state indicates that

$$S_7 : (l + 2, l + 3, l + 4, l + 5, l + 6, +7) \longmapsto (b, a, d, c, f, e).$$

Generally, each element of $S_0(\{0, \ldots, N-1\} \setminus \{1, 2, 4\})$ is shifted to the right alternately by 2 or by 4 positions. Let z_i denotes an output of RC4 in the round i. Then

$$z_2 = S_2(b + 3),\ z_3 = S_3(a - 1),\ z_4 = S_4(d + 3),$$
$$z_5 = S_5(c - 1),\ z_6 = S_6(f + 3),\ z_7 = S_7(e - 1).$$

Let us have $a = 1$ (this is just for illustration, other cases can be handled similarly). Then $z_3 = S_3(a-1) = S_3(0)$. Since we have assumed $0 \leq l < N-10$ it is $z_3 = S_0(0)$.

After N rounds of PRGA, it is again $i = l$ and values of S_N are shifted to the right (by 2 or by 4 positions) compared with S_0. Since $S_0(l + 5) = a$ and N is even, the value a is each N steps shifted to the right by 2 positions. Consequently it is

$$z_{3+N-2} = S_{3+N-2}(a - 1) = S_{3+N-2}(0) = S_0(2).$$

So after approximately $N \cdot N/2$ round we obtain $S_0(k)$ for all even $0 \leq k < N$, $k \notin \{2, 4\}$. In a similar manner we can obtain the second half of the permutation S_0 (use an even b instead of odd $a = 1$).

References

[MOV] Alfred Menezes, Paul C. van Oorschot, Scott A. Vanstone: Handbook of Applies Cryptography, *CRC Press, 1996.*

[Fi94] Hal Finney: An RC4 cycle that can't happen, *September 1994.*

[Ro95] Andrew Roos: A class of weak keys in the RC4 stream cipher. *Posted to sci.crypt, Sept. 1995.*

[Go97] Jovan Dj. Golic: Linear Statistical Weakness of Alleged RC4 Keystream Generator. *Advances in Cryptology—Eurocrypt '97, Lecture Notes of Computer Science, Springer-Verlag, 1997.*

[KMP98] Lars Knudsen, Willi Meier, Bart Preneel, Vincent Rijmen, Sven Verdoolaege: Analysis Methods for (Alleged) RC4. *Advances in Cryptology—Asiacrypt '98, Lecture Notes if Computes Science, Springer-Verlag, 1998.*

[MT99] Serge Mister, S.E. Tavares: Cryptanalysis of RC4-like Ciphers. *Selected Areas of Cryptography (SAC '98), Lecture Notes of Computer Science, Springer-Verlag, 1999.*

[FM00] Scott R. Fluhrer. David A. McGrew: Statistical Analysis of the Alleged RC4 Keystream Generator, *Fast Software Encryption (FSE 2000), Lecture Notes in Computes Science, Springer-Verlag, 2000.*

[GW00] Alexander L. Grosul, Dan S. Wallach: A Related-Key Cryptanalysis of RC4, *Technical Report TR-00-358, Department of Computer Science, Rice University, 2000.*

[FMS01] Scott R. Fluhrer, Itsik Mantin, Adi Shamir: Weaknesses in the Key Scheduling Algorithm of RC4, *Selected Areas of Cryptography (SAC 2001), Lecture Notes of Computer Science, Springer-Verlag, 2001.*

[SAR01] Adam Stubblefield, John Ioannidis, Aviel D. Rubin: Using the Fluhrer Mantin and Shamir Attack to Break WEP, *Technical Report TD-4ZCPZZ, AT&T Labs, 2001.*

[MS01] Itsik Mantin, Adi Shamir (2001): A Practical Attack on Broadcast RC4, *FSE: Fast Software Encryption, FSE'2001, Springer-Verlag, 2001.*

[RSA01] Ron Rivest: RSA security response to weaknesses is key scheduling algorithm of RC4, *Tech. note, RSA Data Security Inc., 2001.*

[Ma01] Itsik Mantin: Analysis of the Stream Cipher RC4, Master Thesis, The Weizmann Institute of Science, Israel, 2001.

[Mi02] Ilya Mironov: (Not So) Random Shuffles of RC4, *Crypto 2002, Lecture Notes of Computer Science, Springer-Verlag, 2002.*

[SP03] Paul Souradyuti, Bart Preneel: Analysis of Non-fortuitous Predictive States of the RC4 Keystream Generator, *Progress in Cryptology—INDOCRYPT 2003: 4th International Conference on Cryptology in India, 2003.*

[SP04a] Paul Souradyuti, Bart Preneel: A New Weakness in the RC4 Keystream Generator and an approach to Improve the Security of the Cipher, *Fast Software Encryption, FSE'04, Springer-Verlag, 2004.*

[Wu05] Hongjun Wu: The Misuse of RC4 in Microsoft Work and Excel, *Technical Report, Institute for Infocomm Research Singapore, 2005.*

[BGN05] Eli Biham, Louis Granboulan, Phong Q. Nguyen: Impossible Fault Analysis of RC4 and Differential Fault Analysis of RC4, *Fast Software Encryption, FSE'05, Springer-Verlag, 2005.*

[Ma05] Itsik Mantin: Predicting and Distinguishing Attacks on RC4 Keystream Generator, *Advances in Cryptology—EUROCRYPT '05, Lecture Notes of Computer Science, Springer-Verlag, 2005.*