ISSN: 1401-5617

Joint Sequences and Factorizations in Free Monoids

Mats Oldin

Research Reports in Mathematics
Number 9, 2006

Department of Mathematics
Stockholm University

Electronic versions of this document are available at
http://www.math.su.se/reports/2006/9

Date of publication: November 22, 2006
2000 Mathematics Subject Classification: Primary 68R15, Secondary 20M35, 06E99, 03E72.
Keywords: Parikh mapping, factorization, factorization schema, joint sequences, Parikh distance, fuzzy shuffle, DNA, DNA-sequencing, fuzzy sets, fuzzy languages, combinatorics, commutative image, semi-commutative image, Parikh matrix, s emigroup, monoid, alphabet, words.

Postal address:
Department of Mathematics
Stockholm University
S-106 91 Stockholm
Sweden

Electronic addresses:
http://www.math.su.se/
info@math.su.se

Joint Sequences and Factorizations in Free Monoids

With applications to DNA-sequencing

Mats Oldin
Department of Mathematics
Stockholm University
matso@math.su.se

Filosofie licentiatavhandling

Avhandlingen kommer att presenteras vid ett seminarium,
onsdagen den 13 december 2006, kl 10:00 i rum 306, hus 6.

Abstract

This paper describes a mathematical framework for rigorously describing and solving problems regarding DNA sequencing. The main problem regards the reconstruction of a DNA-sequence from several partial descriptions of the sequence. The partial descriptions are modeled by, what we call, *semi-commutative images* of different factorizations of the sequence. It is shown that the information given by multiple semi-commutative images could be represented by a single semi-commutative image obtained by, what is defined as the *join* of the images. This join is an application of a general construction which we define for sequences in a general group. This join is also applied to factorizations and so called *factorization schemata*. It is shown that the join operation makes the set of factorizations and factorization schemata into boolean algebras. In two appendices the reconstruction problem is reformulated in alternative ways. One uses the framework of fuzzy set theory to model inexactness in the partial information. The other formulates the problem as a completion problem of Parikh matrices.

Contents

Acknowledgments

First of all I would like to thank Yishao Zhou for her support during the work with this paper. She has followed the work during all its phases from the initial idea to its final form presented here. She has consistently given me positive feedback and encouraged me to go further with my ideas.

I also would like to thank Victor Ufnarovski, Jörgen Backelin and Jesper Carlström for their interest in my work and for their comments which have helped me make the text more readable.

Thanks also to Jan Johanson, who has encouraged me to keep focus on the mathematics and to avoid letting other duties at the institution absorb my full attention.

Finally I would like to thank my wife Jessica, who through her tremendous support has made it possible for me to finish this work.

1 Introduction

The work presented in this paper originates in an algorithmic problem which is a part of a method of DNA-sequencing which the author worked on for the biotech company *Global Genomics AB*. [1]

To get an intuition for what the mathematics presented here could be used for, we start to describe the biological background of the problem.

The problem arose in the context of DNA-sequencing which is a process in molecular biology with the goal of determining the nucleotide order of a given DNA-sequence. There are several ways to perform this process. In the method which has inspired to this paper we are given several partial descriptions of a sequence and from these we are expected to derive the original sequence. Recall that a DNA-sequence consists of a chain of molecules organized in a double-helix. The molecules, called *nucleotides*, are of four types: *adenine* (abbreviated A), *cytosine* (C), *guanine* (G) and *thymine* (T). The double-helix consists of two “complementary” chains of nucleotides. More specifically, the two chains are bonded to each other in such a way that an A in one chain always bonds to a T on the other chain and vice versa. Similarly a C in one chain is bonded to a G in the other chain. Therefore, it is sufficient to describe the double-chain only by giving the order of the nucleotides in one of the chains (chosen by some convention).

[1] The company *Global Genomics AB* no longer exists but the method is now property of the Canadian company *Genizon*.

E.g. $ACCTAGGTA$ is one such representation, the omitted chain in this example is $TGGATCCAT$. In biological literature the molecules are often denoted by the capital letters A, C, G, T. From now on we, however, will use lower-case letters when they are considered as parts of sequences and capital letters when they are considered as molecules.

We will now illustrate the problem by an example. Let us say that the (yet unknown) sequence w is *aactgaccgtaattc*. The sequence w is measured by a scanning device. The device partitions the sequence into consecutive segments, e.g. $(aact, gaccgt, aatt, c)$. Each segment is scanned and the amount of each of the four molecules A, C, G and T is measured. Let k be the number of segments in one segmentation of the sequence. The measurement of the segments is represented by signal vectors $s_i = (s_{i,A}, s_{i,C}, s_{i,G}, s_{i,T}) \in \mathbb{R}^4$ where $i = 1, \ldots, k$. We assume $s_{i,N} \geq 0$ for $N = A, C, G, T$. The magnitude of e.g. $s_{i,A}$ then gives information about how many A were present in the i:th segment. Typically, $s_{i,A}$ is a measurement of the total radiation of fluorescents attached to the A-nucleotides. We assume that the scanner is so precise that e.g. $s_{i,N} = 4$ means that exactly 4 N-nucleotides were present in the segment i. Then, the sequence of signal vectors would be

$$(2,1,0,1),(1,2,2,1)(2,0,0,2)(0,1,0,0).$$

After that the sequence is segmented in another way. Each of the new segments are scanned in the same way. This process of "re-segmentation" and scanning is repeated a number of times.

In the next measurement let us say that w is segmented as

$$(aa, ctgacc, gtaattc).$$

The signal vectors then are

$$(2,0,0,0),(1,3,1,1),(2,1,1,3).$$

Now, given the two (or more) sequences of signal vectors, our problem is to determine w. The first signal vector $(2,0,0,0)$ in the second measurement sequence shows us that w *must* begin with two a's. If we subtract $(2,0,0,0)$ from the first vector $(2,1,0,1)$ in the first sequence we get $(0,1,0,1)$. This tells us that the next two letters are c and t. The order of these are however still uncertain. So, we see that in the above example it is not possible to determine w uniquely. We need more signal vector sequences in order to do that. One question arises: How many measurements do we have to do in order to determine w?

In this paper a formal method will be described for calculating the set of possible words w. It will be shown that the information contained in the two sequences in the above example could be described by the following single sequence

$$(2,0,0,0)(0,1,0,1)(1,2,1,0)(0,0,1,1)(2,0,0,2)(0,1,0,0).$$

With the terminology of this paper the latter sequence is the *join* of the two previous sequences. In Section 3.3 the concept of *joint sequences* will be discussed for sequences of elements in any group. In order to construct the join of two sequences the concept of *factorization schemata* is used.

To the authors best knowledge the concepts of joint sequences and factorization schemata is not seen in the literature to this date. In Section 4 this concept will be used to solve the problem described above. In Section 5 there will be shown how to choose a segmentation of the word in order to successfully reconstruct the word.

Another twist to the problem enters when we let the measurements contain errors. The set of signal vector sequences then can be inconsistent. In Section 6 some indications are given on how to handle the case of errors in the measurements. Appendix A will use the theory of fuzzy sets to handle the inexact measurements. In Appendix B the problem will be formulated as a matrix completion problem. The type of matrices are of a special kind called *Parikh Matrices*.

2 Preliminaries

We will use the machinery used in formal languages and combinatorics of words to formulate the biological problem described above into a precise mathematical setting. We begin by setting our notation. The notation closely follows [Lot97]. Throughout this paper Σ will denote a finite alphabet. The elements of Σ will be called *letters* and finite sequences of letters will be called *words*. Let Σ^* denote the set of all words over Σ, including the empty word ϵ. Σ^* denotes the free monoid generated by Σ with unity ϵ under the operation of *concatenation* (or *product*) of words. Let Σ^+ denote the subset of Σ^* containing all words but the empty word. Σ^+ then is a semigroup. For a subset $X \subset \Sigma^*$ we will let X^* denote the set of all products of words in X.

The *length* of a word $w = a_1 \cdots a_n \in \Sigma^*$, $a_i \in \Sigma$, is the number of letters that w is a product of. We denote the length $|w| = n$. In particular $|\epsilon| = 0$.

We let Σ^n denote the subset of Σ^* containing all words of length n.

An *ordered alphabet* is an alphabet $\Sigma = \{a_1, \ldots, a_k\}$ which have a total order $<$ between the letters. If we have $a_1 < a_2 < \cdots < a_k$, then we write $\Sigma = \{a_1 < a_2 < \cdots < a_k\}$.

We call a word u a *factor* or a *subword* of w if there exist words $x, y \in \Sigma^*$ such that $w = xuy$. The factor is called *proper* if $u \neq w$. We say that u is a *prefix* of w if there exists a word x such that $w = ux$. Similarly we say that u is a *suffix* of w if there exists a word x such that $w = xu$.

For a subset $A \subset \Sigma$ we denote the number of letters of w that belong to A by $|w|_A$. For a single letter $a \in \Sigma$ we abuse the notation and write $|w|_a$ to denote the number of occurrences of a in w.

Let $\Sigma = \{a_1 < a_2 < \cdots < a_k\}$ be an ordered alphabet. The image of the map

$$\Psi : \Sigma^* \to \mathbb{N}^k \tag{1}$$

$$\Psi(w) = (|w|_{a_1}, \ldots, |w|_{a_k}) \tag{2}$$

is often referred to as the *Parikh image* or *commutative image* of w, see e.g. [CK97].

We will regard Σ^* as a subset of $\langle\Sigma\rangle$, the free group over Σ. In that way expressions like

$$u(u^{-1}x) = x \tag{3}$$

are well defined with $u^{-1} \in \langle\Sigma\rangle$.

3 Factorizations of words

3.1 Introduction

In this section we will define the concept of *factorization schemata.* The schema of a factorization determines the length of each factor in a factorization. This concept will later be used to construct the join of two factorizations. It is natural to let Σ^{**} denote the set of finite sequences of words from Σ^*. Let $u, v \in \Sigma^{**}$ where $u = (u_1, \ldots, u_n)$ and $v = (v_1, \ldots, v_m)$ with $u_i, v_j \in \Sigma^*$, for $i = 1, \ldots, n$ and $j = 1, \ldots, m$. We define the *concatenation* uv of u and v by

$$uv = (u_1, \ldots, u_n, v_1, \ldots, v_m). \tag{4}$$

We let $\emptyset$ denote the empty sequence in Σ^{**} and define $\emptyset u = u\emptyset = u$ for all $u \in \Sigma^{**}$. Concatenation is clearly associative so Σ^{**} is a monoid with unity $\emptyset$. Given a sequence $u = (u_1, \ldots u_n) \in \Sigma^{**}$ we define a mapping $\pi : \Sigma^{**} \to \Sigma^*$

$$\pi(u) = \Pi_{i=1}^{n} u_i. \tag{5}$$

In words, π concatenates the words of the sequence u to form a larger word in Σ^*. The fact that

$$\pi(uv) = \pi(u)\pi(v) \tag{6}$$

makes π a morphism. Consider a set $A = \{s_1, \ldots, s_n\}$ with $s_i \in \Sigma^{**}$. For sets like A we will abuse notation and write

$$\pi(A) = \{\pi(s_i) | i = 1, \ldots, n\} \subset \Sigma^*. \tag{7}$$

Definition 3.1 (Factorization) *Given a word $w \in \Sigma^*$ and a sequence of words $u = (u_1, \ldots, u_n) \in \Sigma^{**}$, we call u a* factorization *of w if $w = \pi(u)$. Let $\mathcal{X} = \{X_i\}$ for $i = 1, \ldots, n$ where $X_i \subset \Sigma^*$. If $u_i \in X_i$ for $i = 1, \ldots, n$ then we say that u is an $\mathcal{X}$*-factorization*. In particular if $u_i \in X$ for all $i = 1, \ldots, n$ and some $X \subset \Sigma^*$ we call the factorization u an X*-factorization*.*

The above definition is an extension of the defintion found e.g. in [Lot02].

Definition 3.2 (Code) *Let $X \subset \Sigma^*$. We say that X is a* code *if the equation*

$$x_1 \cdots x_m = y_1 \cdots y_n \tag{8}$$

implies $m = n$ and $x_i = y_i$ for $i = 1, \ldots, m$ whenever all elements $x_1, \ldots, x_m, y_1, \ldots, y_n \in X$.

That X is a code means that there is no relations between elements in X. One could also describe it as that any word $w \in \Sigma^*$ has at most one X-factorization. The following theorem motivates the name code.

Theorem 3.3 (Codes) *Let Y be any alphabet. A set $X \subset \Sigma^*$ is a code iff any morphism $\phi : Y^* \to \Sigma^*$ induced by a bijection from Y onto X is injective.*

In other words each word in Y^* is coded into a concatenation of "code words" from X. That X is a code means that the original word in Y^* uniquely could be determined by the resulting "code word".

Theorem 3.4 *Let $\phi : Y^* \to \Sigma^*$ be an injective morphism. If $Z \subset Y^+$ is a code, then $\phi(Z)$ is a code. If $B \subset \Sigma^+$ is a code, then $\phi^{-1}(B)$ is a code.*

For proofs of the above theorems see e.g. [Lot02].

Example 3.5 (Codes) *The set $X = \{aa, ab, ac, b, c\}$ is a code over the alphabet $Y = \{a, b, c\}$. Let $\phi : Y^* \to \Sigma^*$ be the morphism induced by $\phi(a) = a$, $\phi(b) = ab$ and $\phi(c) = bb$. According to Theorem 3.4 the set $\phi(X) = \{aa, aab, abb, ab, bb\}$ is a code.*

We now procede with the definition of *minimal factorizations.* This is a concept which we will use in Section 3.3.

Definition 3.6 (Minimal Factorizations) *Let $u = (u_1, \ldots, u_m)$ be an X-factorization of a word $w \in \Sigma^*$ such that $u_i u_{i+1} \cdots u_{i+s} \in X$ is false for all integers i and s where $i = 1, \ldots, m-1$ and $s > 0$ such that $s + i \leq m$. Then we call u a* minimal X-factorization *of w.*

In a minimal factorization no product of consecutive elements, e.g. $u_i u_{i+1}$, belongs to X.

Theorem 3.7 *If X is a code, then any X-factorization of a word $w \in \Sigma^*$ is minimal.*

Proof. Take an X-factorization $u = (u_1, \ldots, u_m)$ of w. For any i and j with $1 \leq i < j \leq m$, define

$$u_{i,j} = u_i u_{i+1} \cdots u_j \tag{9}$$

Then clearly $u_{i,j}$ could not belong to X since this would contradict the properties of equation (8) given in the definition of a code. Therefore $(u_1, \ldots, u_m)$ must be a minimal factorization of u. □

The following example shows us that given a set X, a word can have several different minimal factorizations.

Example 3.8 *Let $w = abcba \in \Sigma^*$ and let $X = \{a, ab, cb, bcb\}$. Then both $(ab \cdot cb \cdot a)$ and $(a \cdot bcb \cdot a)$ are minimal X-factorizations of w.*

Theorem 3.9 *Any X-factorization $u = (u_1, \ldots, u_m)$ of a word w can be transformed into a minimal factorization v of w.*

Proof. If u is not minimal take the smallest integers i and j with $1 \leq i < j \leq m$ such that $u_i \cdots u_j \in X$. Then replace u with

$$u^{(1)} = (u_1, \ldots, u_{i-1}, u_i \cdots u_j, u_{j+1}, \ldots, u_m).$$

If $u^{(1)}$ is not minimal continue this procedure. Since the factorization has a finite number of factors the procedure will terminate after say s steps. Then take $v = u^{(s)}$ which must be minimal. □

3.2 Factorization Schemata

Take any factorization $u = (u_1, \ldots, u_m) \in \Sigma^{**}$ of the word $w \in \Sigma^n$. We can specify the factors $u_1, \ldots, u_m$ of u by specifying the position of the first letter in each factor as it is positioned in w. Given that we know the length of the word $aaba$ the factorization (aa, ba) could be specified by just giving the position of the letter b in the last factor, which in this case is 3. Obviously there is no need to specify the first letter of the first factor, since this factor is completely determined once we know where the second factor starts. For a word of length n the positions of the letters are numbers between 1 and n. Thus, a general factorization could be described by a subset of these numbers where each number represents the start position of one of the factors. But as we noted, the start of the first factor need not be specified so it is sufficient with a subset of the numbers between 2 and n. This fact is the motivation for the following definition.

Definition 3.10 (Factorization Schema) *Let n be a positive integer. We define an n-*factorization schema *as an ordered subset of $\{2, ..., n\}$.*

We will usually regard a factorization schema as an ordered set with the usual order of natural numbers. We denote the set of n-factorization schemata by $\mathcal{F}_n$. In particular we denote the set $\{2, \ldots, n\}$ by λ. Given $F \in \mathcal{F}_n$ one can construct a factorization on the words in Σ^n by splitting the word at the indices in F.

Definition 3.11 (Induced Factorization) *Let F be an n-factorization schema and $w = a_1 \cdots a_n \in \Sigma^*$ a non-empty word of length n with $a_i \in \Sigma$. Let $k_1 < \ldots < k_s$ be the elements of F given in order and set $k_0 = 1$ and $k_{s+1} = n + 1$. We define the factorization of w* induced *by F as*

$$\overline{F}(w) = (w_1, \ldots, w_{s+1})$$

where

$$w_i = a_{k_{i-1}} \cdots a_{k_i - 1} \tag{10}$$

*for $i = 1, \ldots, s + 1$. For $F = \emptyset$ we define $\overline{F}(w) = (w)$. We call $\overline{F}$ the n-*factorization map *induced by F. For the particular case $F = \{2, \ldots, n\} = \lambda$ we denote $\overline{F}$ by $\overline{\lambda}$.*

Theorem 3.12 *Let F and G be two n-factorization schemata. Then $\overline{F} = \overline{G}$ iff $F = G$.*

Proof. If $F = G$ then $\overline{F} = \overline{G}$ by definition. Assume $\overline{F} = \overline{G}$. Then for any $w \in \Sigma^n$

$$\overline{F}(w) = \overline{G}(w) = (w_1, \ldots, w_m)$$

for some fixed integer m. Let $f(w_i)$ be the position of the first letter in w_i in the word w for $i = 2, \ldots, m$. Then

$$F = G = \{f(w_i) | i = 2, \ldots, m\}$$

□

Example 3.13 *Let $\Sigma = \{a, b, c\}$ and $w = aabcabcb$. For the sake of clarity we rename the letters of w so that $w = a_1a_2a_3a_4a_5a_6a_7a_8$. Let $F = \{3, 5, 8\}$. Then*

$$\begin{array}{rcccc} w_1 = a_1 \cdots a_{3-1} & = & a_1a_2 & = & aa \\ w_2 = a_3 \cdots a_{5-1} & = & a_3a_4 & = & bc \\ w_3 = a_5 \cdots a_{8-1} & = & a_5a_6a_7 & = & abc \\ w_4 = a_8 \cdots a_{9-1} & = & a_8 & = & b. \end{array}$$

This gives

$$\overline{F}(w) = (w_1, w_2, w_3, w_4) = (aa, bc, abc, b).$$

We also have

$$\overline{\lambda}(w) = (a, a, b, c, a, b, c, b).$$

We note that if $|F| = s$ then $\overline{F}(w)$ has $s+1$ factors. For a word $w \in \Sigma^n$ we let $\mathcal{F}_n(w)$ denote the set of factorizations of w

$$\mathcal{F}_n(w) = \{\overline{F}(w) | F \in \mathcal{F}_n\}. \tag{11}$$

Clearly

$$|\mathcal{F}_n(w)| = 2^{n-1}. \tag{12}$$

Given a factorization $w = (w_1, \ldots, w_{s+1})$ a corresponding factorization schema $F \in \mathcal{F}_n$ could be constructed. It is given by $F = \{k_1, \ldots, k_s\}$ where the k_i are easily calculated by equation (10) in Definition 3.11.

Let $\mathcal{F}_n(\Sigma^n)$ be the set of factorizations of all words of length n. For each n-factorization schemata F, the induced n-factorization map $\overline{F}$ is an injective function $\Sigma^n \to \Sigma^{**}$. The inverse of $\overline{F}$ is clearly the map π defined in (5). Take for example $w \in \Sigma^n$ and $\overline{F}(w) = (w_1, \ldots, w_s) \in \mathcal{F}_n(\Sigma^n)$. Then

$$\pi(\overline{F}(w)) = \pi((w_1, \ldots, w_s)) = w. \tag{13}$$

We let $\overline{\mathcal{F}}_n$ denote the set of n-factorization maps

$$\overline{\mathcal{F}}_n = \{\overline{F} | F \in \mathcal{F}_n\}. \tag{14}$$

3.3 Joint Sequences

We will now describe a general construction of how to "join" two sequences as described in the introduction to this paper. The construction is applicable to all sequences in groups with one constraint: The "partial products" as defined below must be increasing.

Let M be a group with identity $\mathtt{1}$ and a norm $|\cdot| : M \to \mathbb{R}$ with the following properties

$$\begin{array}{lccc} \forall a \in M: & |a| = 0 & \Leftrightarrow & a = \mathtt{1}, \\ \forall a \in M: & |a| & = & |a^{-1}|, \\ \forall a, b \in M: & |ab| & \leq & |a| + |b|. \end{array} \tag{15}$$

Denote the set of finite sequences of elements of M by M^*. Now, consider two sequences U and V from M^*. Let $U = (U_1, \ldots, U_s) \in M^*$ and $V = (V_1, \ldots, V_t) \in M^*$ for some positive integers s and t. Define the "partial products" $U'_i = U_1 \cdots U_i$ and $V'_j = V_1 \cdots V_j$ for $i = 1, \ldots, s$ and $j = 1, \ldots, t$. Then since M is a group $U'_i, V'_j \in M$.

We now assume that U and V are given such that $|U'_i| < |U'_{i+1}|$ and $|V'_j| < |V'_{j+1}|$, i.e. U'_i and V'_j are increasing sequences in M.

We proceed by constructing the sets $U' = \{U'_i\}_{i=1}^s$ and $V' = \{V'_i\}_{i=1}^t$. It is now possible to define an order $\prec_{U,V}$ on $U' \cup V'$ by

$$x \prec_{U,V} y \quad \text{if} \quad \left\{ \begin{array}{c} |x| < |y| \\ \text{or} \\ |x| = |y| \text{ and } x \in U' \text{ and } y \in V' \end{array} \right.$$

The assumption that the series of partial products is increasing makes the order total. We see this since if $x, y \in U'$ or $x, y \in V'$, then by this assumption on U and V, we have $x \neq y$ which implies either $|x| < |y|$ or $|y| < |x|$. Let $\{x_1 \prec_{U,V} \cdots \prec_{U,V} x_{s+t}\}$ be the ordered set of elements of $U' \cup V'$. Define

$$\varphi_{U,V} : \{1, \ldots, s+t\} \to U' \cup V' \tag{16}$$

such that $\varphi_{U,V}(i) = x_i$. An obvious property of $\varphi_{U,V}$ is that $|\varphi_{U,V}(i)| \leq |\varphi_{U,V}(j)|$ for $i \leq j$. We construct a new sequence Z in M in the following way. Let

$$Z'_1 \;=\; \varphi_{U,V}(1) \tag{17}$$

$$Z'_i \;=\; \varphi_{U,V}(i-1)^{-1} \cdot \varphi_{U,V}(i) \tag{18}$$

for $i = 2, \ldots, s+t$ where $\varphi_{U,V}(i-1)^{-1}$ is the inverse of $\varphi_{U,V}(i-1)$ in M. Let $Z = Z_1, \ldots, Z_r$, $r \leq s+t$ be the subsequence of Z' where all unity elements have been removed. We call Z for the *joint sequence* of U and V with respect to the norm $|\cdot|$. We assume that the norm used is understood from the context and denote the joint sequence by $Z = U \vee V$. Let S be a subset of M. If $Z_i \in S$ for all i then we say that U and V are *S-joinable.*

Example 3.14 *Consider the sequences $U = (1, 3, 5)$ and $V = (2, 6)$ in $\mathbb{Z}$. Using the above construction we will now construct $U \vee V$. We* *use addition as binary operation and absolute value as norm. We then obtain the following partial sums*

$$\begin{array}{rclcrcl} U'_1 & = & 1 & \quad & V'_1 & = & 2 \\ U'_2 & = & 4 & & V'_2 & = & 8 \\ U'_3 & = & 9. & & & & \end{array}$$

We see that the partial sums are increasing so it is possible to use our join construction on these two sequences. By ordering these numbers by $\prec_{U,V}$ *we get*

$$U'_1 \prec_{U,V} V'_1 \prec_{U,V} U'_2 \prec_{U,V} V'_2 \prec_{U,V} U'_3.$$

Equations (17) and (18) then give us

$$\begin{array}{rclclcl} Z'_1 & = & U'_1 & = & 1 & & \\ Z'_2 & = & V'_1 - U'_1 & = & 2-1 & = & 1 \\ Z'_3 & = & U'_2 - V'_1 & = & 4-2 & = & 2 \\ Z'_4 & = & V'_2 - U'_2 & = & 8-4 & = & 4 \\ Z'_5 & = & U'_3 - V'_2 & = & 9-8 & = & 1 \end{array}$$

Since the sequence Z'_i *does not contain any zeroes we get*

$$(1,3,5) \vee (2,6) = (1,1,2,4,1)$$

Example 3.15 *Let* $\Sigma = \{a,b\}$*. Take two sequences* $U = (aa, ba, a)$ *and* $V = (a, aba, a)$ *in* $\langle\Sigma\rangle$*. We will now construct the join of these two sequences. We will use the binary operation of concatenation from* $\langle\Sigma\rangle$ *and use word length as norm. The partial products are*

$$\begin{array}{rclcrcl} U'_1 & = & aa & \quad & V'_1 & = & a \\ U'_2 & = & aaba & & V'_2 & = & aaba \\ U'_3 & = & aabaa & & V'_3 & = & aabaa \end{array}$$

The partial products are increasing so it is possible to use our join on the sequences. It is obvious that the partial products must be increasing for any sequence of words in Σ^* *for any given alphabet* Σ*. We order the words* U'_i *and* V'_i *for* $i = 1, \ldots, 3$ *using* $\prec_{U,V}$ *and we get*

$$V'_1 \prec_{U,V} U'_1 \prec_{U,V} U'_2 \prec_{U,V} V'_2 \prec_{U,V} U'_3 \prec_{U,V} V'_3$$

The next step in the construction results in

$$\begin{array}{rclclcl} Z'_1 & = & V'_1 & = & a & & \\ Z'_2 & = & (V'_1)^{-1}U'_1 & = & (a)^{-1}aa & = & a \\ Z'_3 & = & (U'_1)^{-1}U'_2 & = & (aa)^{-1}aaba & = & ba \\ Z'_4 & = & (U'_2)^{-1}V'_2 & = & (aaba)^{-1}aaba & = & \epsilon \\ Z'_5 & = & (V'_2)^{-1}U'_3 & = & (aaba)^{-1}aabaa & = & a \\ Z'_5 & = & (U'_3)^{-1}V'_3 & = & (aabaa)^{-1}aabaa & = & \epsilon \end{array}$$

After removal of the unity elements, i.e. of the empty words, we get

$$(aa, ba, a) \vee (a, aba, a) = (a, a, ba, a).$$

3.4 The Boolean Algebra of Factorizations

In the previous section we defined joins of sequences in a general group. In this section we will be more specific and look at joins of sequences in $\langle\Sigma\rangle$. The elements of these sequences are regarded as factors in a factorization. We will also define a construction of joins of factorization schemata. Then we will show that this notion of join in a natural way coincide with the previous notion when applied to the induced factorizations in $\langle\Sigma\rangle$.

Definition 3.16 (Joins and Meets) *Let $F, G \in \mathcal{F}_n$ be two n-factorization schemata. We will define the* join $\overline{F} \vee \overline{G}$ *of $\overline{F}$ and $\overline{G}$ as the n-factorization map induced by $F \cup G$. We define the* meet $\overline{F} \wedge \overline{G}$ *of $\overline{F}$ and $\overline{G}$ as the n-factorization map induced by $F \cap G$. That is*

$$\begin{aligned} \overline{F} \wedge \overline{G} &= \overline{F \cap G}, \qquad (19) \\ \overline{F} \vee \overline{G} &= \overline{F \cup G}. \qquad (20) \end{aligned}$$

Example 3.17 *As in Example 3.13 let $\Sigma = \{a, b, c\}$, $w = aabcabcb$, $F = \{3, 5, 8\}$ and also let $G = \{4, 5, 7\}$. Then $F \cup G = \{3, 4, 5, 7, 8\}$ and $F \cap G = \{5\}$ so*

$$\begin{aligned} \overline{F}(w) &= (aa, bc, abc, b), \\ \overline{G}(w) &= (aab, c, ab, cb), \\ \overline{F} \vee \overline{G}(w) &= (aa, b, c, ab, c, b), \\ \overline{F} \wedge \overline{G}(w) &= (aabc, abcb). \end{aligned}$$

It is natural to look at a join of two factorization maps as a refinement of the factorizations. On the other hand, taking the meet of two factorization maps gives a more coarse factorization of the word.

Definition 3.18 *If $\overline{F} = \overline{F} \wedge \overline{G}$ then we write $\overline{F} \leq \overline{G}$.*

Theorem 3.19 *$\overline{F} \leq \overline{G}$ iff $F \subset G$.*

Proof. By definition $\overline{F} = \overline{F} \wedge \overline{G} = \overline{F \cap G}$ which by Theorem 3.12 is equivalent to $F = F \cap G$ so we have $F \subset G$ and the theorem follows. □

We will now show that two factorizations of a word w can be considered as two sequences that must be joinable. We will also investigate the connection between their join and the factorization of w obtained by the join of the factorizations corresponding factorization schemata. We will actually show that these two factorizations coincide. We state this as follows.

Theorem 3.20 *Let $F, G \in \mathcal{F}_n$ and $w \in \Sigma^n$. $\overline{F}(w)$ and $\overline{G}(w)$ considered as sequences in the free group $\langle\Sigma\rangle$ together with the word length $|\cdot|$ as norm admits the construction of a joint sequence $\overline{F}(w) \vee \overline{G}(w)$ in $\langle\Sigma\rangle$ as described in Section 3.3. Moreover*

$$\overline{F}(w) \vee \overline{G}(w) = (\overline{F} \vee \overline{G})(w) \tag{21}$$

Remark 3.21 *The main part of the proof of this theorem consists in the actual construction of the join $\overline{F}(w) \vee \overline{G}(w)$. In the proof we will introduce an auxiliary function φ. In the remaining part of this paper we sometimes need to refer to this function along with some other elements of the proof (for instance in Corollary 3.22).*

Proof. Assume $w = a_1 \cdots a_n$ for $a_i \in \Sigma$ and assume $F = (k_1, \ldots, k_s)$ and $G = (k'_1, \ldots, k'_t)$. Let $h_1 \leq h_2 \leq \cdots \leq h_p$ be the elements of $F \cup G$ given in order and let $h_0 = 1$ and $h_{p+1} = n + 1$. Then

$$(\overline{F} \vee \overline{G})(w) = (w_1, w_2, \ldots, w_{p+1}) \tag{22}$$

where

$$w_i = a_{h_{i-1}} \cdots a_{h_i - 1} \tag{23}$$

for $i = 1, \ldots, p + 1$.

We now turn to the construction of $\overline{F}(w) \vee \overline{G}(w)$. Let $\overline{F}(w) = (u_1, \ldots, u_{s+1})$ and $\overline{G}(w) = (v_1, \ldots, v_{t+1})$ where

$$u_i = a_{k_{i-1}} \cdots a_{k_i - 1} \quad \text{and} \quad v_j = a_{k'_{j-1}} \cdots a_{k'_j - 1} \tag{24}$$

for $i = 1, \ldots, s + 1$ and $j = 1, \ldots, t + 1$ and where $k_0 = k'_0 = 1$ and $k_{s+1} = k_{t+1} = n + 1$. Then

$$\begin{aligned} F'_i &= u_1 \cdots u_i = a_1 \cdots a_{k_i - 1} & (25) \\ G'_j &= v_1 \cdots v_j = a_1 \cdots a_{k'_j - 1} & (26) \end{aligned}$$

which shows that $|F'_i| = k_i - 1$ and $|G'_j| = k'_j - 1$. We let the function $\varphi_{F,G}$ order the set $\{F'_i\}_{i=1}^{s+1} \cup \{G'_j\}_{j=1}^{t+1}$ according to word length, as shown in Section 3.3. So, by taking word length, the function $|\varphi_{F,G}|$ gives the elements in the set $\{k_i - 1\}_{i=1}^{s+1} \cup \{k'_j - 1\}_{j=1}^{t+1}$ in order (since these numbers are the corresponding word lengths by (25) and (26)).

Now $h_{i'}$ also gives the elements in $\{k_i\}_{i=1}^{s+1} \cup \{k'_j\}_{j=1}^{t+1}$ in order so we have

$$|\varphi_{F,G}(i')| = h_{i'} - 1 \tag{27}$$

for $i' = 1, \ldots, p+1$. This gives us

$$\varphi_{F,G}(i') = a_1 \cdots a_{h_{i'}-1} \tag{28}$$

and

$$\begin{aligned} Z_1 &= \varphi_{F,G}(1) = a_1 \cdots a_{h_1-1} = w_1 & (29) \\ Z_{i'} &= \varphi_{F,G}(i'-1)^{-1}\varphi_{F,G}(i') \\ &= (a_1 \cdots a_{h_{i'_{-1}}-1})^{-1}(a_1 \cdots a_{h'_i-1}) \\ &= a_{h_{i'_{-1}}} \cdots a_{h_{i'}-1} \\ &= w_{i'} & (30) \end{aligned}$$

for $i' = 2, \ldots, p+1$ where the last equality follows from (23). We have shown the existens of $\overline{F}(w) \vee \overline{G}(w) = (Z_1, \ldots, Z_{p+1})$. Equations (22), (29) and (30) further show that $\overline{F}(w) \vee \overline{G}(w) = (\bar{F} \vee \bar{G})(w)$. □

Corollary 3.22 *In the proof of the above theorem $\varphi(i-1)$ is a prefix of $\varphi(i)$ for $i = 2, \ldots, p+1$.*

Proof. This is directly seen by equation (28). □

Clearly the set $\mathcal{F}_n$ is closed under $\vee$ and $\wedge$. These operations make $\mathcal{F}_n$ a lattice. The maximal element of $\mathcal{F}_n$ is $\overline{M}$ where $M = \{2, \ldots, n\}$. We denote this factorization schema by λ. The minimal element is $\overline{\emptyset}$ which we denote by 0.

Theorem 3.23 *The operations $\vee$ and $\wedge$ are associative and commutative on elements in $\mathcal{F}_n$.*

Proof. This follows directly from the associativity and commutativity of $\cup$ and $\cap$. □

Theorem 3.24 *Let $F, G, H \in \mathcal{F}_n$. Then the distributive laws hold*

$$\begin{aligned} \overline{F} \vee (\overline{G} \wedge \overline{H}) &= (\overline{F} \vee \overline{G}) \wedge (\overline{F} \vee \overline{H}), & (31) \\ \overline{F} \wedge (\overline{G} \vee \overline{H}) &= (\overline{F} \wedge \overline{G}) \vee (\overline{F} \wedge \overline{H}). & (32) \end{aligned}$$

Proof. This follows directly from the distributive laws of $\cup$ and $\cap$. □

Theorem 3.25 *Let $F \in \mathcal{F}_n$ and let $G = \{2, \ldots, n\} \setminus F$. We then define $\overline{F}^{-1} = \overline{G}$. Then the following equations hold*

$$\begin{aligned}
\overline{F} \wedge 0 &= 0 && (33)\\
\overline{F} \vee 0 &= \overline{F} && (34)\\
\overline{F} \wedge \lambda &= \overline{F} && (35)\\
\overline{F} \vee \lambda &= \lambda && (36)\\
\overline{F} \vee \overline{F}^{-1} &= \lambda && (37)\\
\overline{F} \wedge \overline{F}^{-1} &= 0 && (38)\\
\overline{F} \wedge \overline{F} = \overline{F} \vee \overline{F} &= \overline{F}. && (39)
\end{aligned}$$

Proof. All these relations are easily verified by simple computation. □

Corollary 3.26 *The structure $(\mathcal{F}_n, \vee, \wedge, \lambda, 0)$ is a boolean algebra.*

Proof. This follows directly from the properties in Theorem 3.25. □

4 Semi-Commutative Images

A word in Σ^* is a non-commutative product of letters in Σ. In this section we will study *commutative images* of words. The commutative image of a word is the product of the same letters but the product is commutative. We will also look at commutative images of factorizations. In this case the order of the factors in the factorization is preserved but the factors are replaced by their commutative images. Thus some non-commutativity is kept. We have chosen to call this a *semi-commutative image.*

4.1 Commutative Images

For some positive integer m let $\mathbb{N}^m$ be the monoid of m-dimensional vectors over $\mathbb{N}$ with ordinary vector addition as binary operation. Let $\bar{0} = (0, \ldots, 0) \in \mathbb{N}^m$. Let $\Sigma = \{a_1, \ldots, a_m\}$ be an alphabet. We define a function $\Psi : \Sigma^* \to \mathbb{N}^m$ by

$$\Psi(w) = (|w|_{a_1}, \ldots, |w|_{a_m}). \tag{40}$$

In particular $\Psi(\epsilon) = \bar{0}$. Recall that $|w|_{a_i}$ denotes the number of occurrences of a_i in w. Observe that under vector addition we have

$$\Psi(w_1 w_2) = \Psi(w_1) + \Psi(w_2). \tag{41}$$

Ψ is easily seen to be an epimorphism between Σ^* and $\mathbb{N}^m$. The image $\Psi(w)$ is referred to as the *commutative image* (or *Parikh vector*) of w.

The morphism Ψ is sometimes referred to as the *Parikh mapping*. See e.g. [Lot97].

Example 4.1 *With $w = aabcabcb$ we have*

$$\Psi(w) = (3, 3, 2).$$

The term commutative image refers to the fact that if Σ^* were commutative the word $w = aabcabcb$ could be written $a^3b^3c^2$ and would be uniquely determined by the image $\Psi(w)$.

We will also use the following extension of Ψ. Let as before $\langle\Sigma\rangle$ be the free group generated by Σ. Let $a \in \Sigma$. Then for $a^{-1} \in \langle\Sigma\rangle$ we define the homomorphism

$$\Psi(a^{-1}) = -\Psi(a). \tag{42}$$

With this extension Ψ becomes an epimorphism between $\langle\Sigma\rangle$ and $\mathbb{Z}^m$.

We now consider the inverse of Ψ. Since several words in Σ^* will have the same commutative image, the inverse of Ψ maps commutative images to subsets of Σ^*. Let $\mathcal{P}(\Sigma^*)$ be the set of subsets of Σ^*. We define the inverse of Ψ as a function from $\mathbb{N}^m$ to $\mathcal{P}(\Sigma^*)$. For $U \in \mathbb{N}^m$ we define

$$\Psi^{-1}(U) = \{w \in \Sigma^* | \Psi(w) = U\}. \tag{43}$$

$\Psi^{-1}(U)$ is the set of all words with the same commutative image U.

Example 4.2 *With $w = aabcabcb$ and $w' = bacacabb$ we have*

$$\Psi(w) = \Psi(w') = (3, 3, 2)$$

so w and w' both belongs to $\Psi^{-1}(3, 3, 2)$.

For $U = (u_1, \ldots, u_s)$ the cardinality of $\Psi^{-1}(U)$ is given by

$$|\Psi^{-1}(U)| = \frac{(\sum_{i=1}^{s} u_i)!}{\prod_{i=1}^{s}(u_i!)}. \tag{44}$$

4.2 Sequences of Commutative Images

One of the main topics in this paper is factorizations. In this section we will discuss the concept of commutative images of factorizations. The construction we will use preserves the order of the factorization and is therefore called the *semi-commutative image* of the factorization.

We extend Ψ to a mapping Ψ^* between sequences in Σ^* and sequences in $\mathbb{N}^m$. We will denote the set of finite sequences of vectors in $\mathbb{N}^m$ by $\mathbb{N}^{m*}$. Thus Ψ^* is a function $\Psi^* : \Sigma^{**} \to \mathbb{N}^{m*}$ which we define as follows. For $s = (w_1, \ldots, w_s) \in \Sigma^{**}$ we let

$$\Psi^*(s) = (\Psi(w_1), \ldots, \Psi(w_s)). \tag{45}$$

Now we look at the inverse of Ψ^*. Let $\mathcal{P}(\Sigma^*)^*$ be the set of sequences of subsets of Σ^*. We define the inverse of Ψ^* as a function $\Psi^{*-1} : \mathbb{N}^{m*} \to \mathcal{P}(\Sigma^*)^*$. [2] For $U = (U_1, \ldots, U_s) \in \mathbb{N}^{m*}$ we have

$$\Psi^{*-1}(U) = (\Psi^{-1}(U_1), \ldots, \Psi^{-1}(U_s)). \tag{46}$$

The cardinality of $\Psi^{*-1}(U)$ is easily computed by using (44)

$$|\Psi^{*-1}(U)| = \prod_{i=1}^{s} |\Psi^{-1}(U_i)|. \tag{47}$$

4.3 Induced Semi-Commutative Images

Each n-factorization schemata $F \in \mathcal{F}_n$ lets us define a function $\Psi^*_{\overline{F}} : \Sigma^n \to \mathbb{N}^{m*}$ by

$$\Psi^*_{\overline{F}}(w) = (\Psi^* \circ \overline{F})(w). \tag{48}$$

The above function maps a word to a semi-commutative image where the sequence elements are determined by a factorization.

Again several words in Σ^n have the same semi-commutative image. Hence, the inverse of $\Psi^*_{\overline{F}}$ is a map to subsets of Σ^n. More precisely, the inverse of $\Psi^*_{\overline{F}}$ is a function $\mathbb{N}^{m*} \to \mathcal{P}(\Sigma^n)$ defined by

$$(\Psi^*_{\overline{F}})^{-1} = \pi \circ \Psi^{*-1}, \tag{49}$$

where $\pi(A)$ for $A \subset \Sigma^{n*}$ denotes the set $\{\pi(u) | u \in A\}$, cf. the definition of π in (5). Note the difference between Ψ^{*-1} and $\Psi^{*-1}_{\overline{F}}$. The function Ψ^{*-1} maps sequences to sequences of sets while $\Psi^{*-1}_{\overline{F}}$ maps sequences to sets of words.

4.4 Commutative Closures

Given a word we will need the following convenient way of denoting the set of all other words with the same commutative or semi-commutative image.

Definition 4.3 (Commutative Closure) *Let $w \in \Sigma^n$ and $F \in \mathcal{F}_n$. We introduce the following notation*

$$\begin{aligned} \mathcal{C}(w) &= \Psi^{-1}(\Psi(w)) && (50) \\ \mathcal{S}_{\overline{F}}(w) &= \Psi^{*-1}_{\overline{F}}(\Psi^*_{\overline{F}}(w)) && (51) \end{aligned}$$

$\mathcal{C}(w) \subset \Sigma^n$ *is the* commutative closure *of w. $\mathcal{S}_{\overline{F}}(w) \subset \Sigma^n$ is the* semi-commutative closure *of w* induced by F. *In particular $\mathcal{C}(w) = \mathcal{S}_{\overline{\emptyset}}(w)$.*

[2] One could also define $\Psi^{*-1} : \mathbb{N}^{m*} \to \mathcal{P}(\Sigma^{**})$ with $U = (U_1, \ldots, U_s) \mapsto \{(w_1, \ldots, w_s) \in \Sigma^{**} | \Psi(w_i) = U_i\}$. But then the structure of U would be lost.

Example 4.4 *With w and w' as in Example 4.2 we have $w' \in \mathcal{C}(w)$.*

Theorem 4.5 *Let $w \in \Sigma^n$ and $F, G \in \mathcal{F}_n$ such that $\overline{F} \leq \overline{G}$. Then the following hold*

$$\mathcal{S}_{\overline{G}}(w) \subset \mathcal{S}_{\overline{F}}(w). \tag{52}$$

Proof. Let $x \in \mathcal{S}_{\overline{G}}(w)$. Assume that $F = \{k_1 < \cdots < k_s\}$ and $G = \{k'_1 < \cdots < k_t\}$. Let

$$\begin{aligned} \overline{F}(x) &= (x_1, \ldots, x_{s+1}) && (53)\\ \overline{F}(w) &= (w_1, \ldots, w_{s+1}) && (54)\\ \overline{G}(x) &= (x'_1, \ldots, x'_{t+1}) && (55)\\ \overline{G}(w) &= (w'_1, \ldots, w'_{t+1}). && (56) \end{aligned}$$

The assumption $x \in \mathcal{S}_{\overline{G}}(w)$ says that $\Psi^*_{\overline{G}}(x) = \Psi^*_{\overline{G}}(w)$ and thus that $\Psi(x'_j) = \Psi(w'_j)$ for $j = 1, \ldots, t+1$. We want to prove $\Psi(x_i) = \Psi(w_i)$ for $i = 1, \ldots, s+1$. Now $\overline{F} \leq \overline{G}$ implies $F \subset G$ by Theorem 3.19 so for all $i = 1, \ldots, s$ there exists a $j_i \in \{1, \ldots, t\}$ such that $k_i = k'_{j_i}$. Therefore the first position of x_i in x is the same as the first position in $x'_{k'_{j_i}}$. We set $k'_{j_1} = 1$ and $k'_{j_{s+2}} = t+2$ and we get the following relations

$$\begin{aligned} x_i &= x'_{k'_{j_i}} \cdots x'_{k'_{j_{i+1}}-1} && (57)\\ w_i &= w'_{k'_{j_i}} \cdots w'_{k'_{j_{i+1}}-1} && (58) \end{aligned}$$

for $i = 1, \ldots, s+1$. The wanted equalities follows from

$$\begin{aligned} \Psi(x_i) &= \Psi(x'_{k'_{j_i}} \cdots x'_{k'_{j_{i+1}}-1})\\ &= \Psi(x'_{k'_{j_i}}) + \cdots + \Psi(x'_{k'_{j_{i+1}}-1})\\ &= \Psi(w'_{k'_{j_i}}) + \cdots + \Psi(w'_{k'_{j_{i+1}}-1})\\ &= \Psi(w'_{k'_{j_i}} \cdots w'_{k'_{j_{i+1}}-1})\\ &= \Psi(w_i). \end{aligned}$$

This shows that $\Psi_{\overline{F}}(x) = \Psi_{\overline{F}}(w)$ so that $x \in \mathcal{S}_{\overline{F}}(w)$.

□

Corollary 4.6 *Let $w \in \Sigma^n$ and $F, G \in \mathcal{F}_n$ such that $\overline{F} \leq \overline{G}$. Then the following chain of inclusions is valid*

$$\{w\} = \mathcal{S}_{\overline{\lambda}}(w) \subset \mathcal{S}_{\overline{G}}(w) \subset \mathcal{S}_{\overline{F}}(w) \subset \mathcal{S}_{\overline{0}}(w) = \mathcal{C}(w) \subset \Sigma^n. \tag{59}$$

4.5 Factorization Schemata Induced by Semi-Commutative Images

In Section 4.3 we showed how a factorization schema could be used to construct a sequence of commutative images. We will now investigate how a factorization schema is induced by a sequence of commutative images.

Let Σ be a finite alphabet. Let $m = |\Sigma|$ and $M = \mathbb{N}^m$. Take a sequence $U \in M^{s+1}$ with $U = (U_1, \ldots, U_{s+1})$. For $U_i \in \mathbb{N}^m$ let $|U_i|$ denote the sum the elements in U_i and assume $n = \sum_{i=1}^{s+1} |U_i|$ and $|U_i| \neq 0$ for $i = 1, \ldots, s+1$. We will regard U as a semi-commutative image and now show how to construct an induced factorization schema from U.

Let $k_i = (\sum_{j=1}^{i} |U_j|) + 1$ for $i = 1, \ldots, s$. Clearly the sequence $k_1, \ldots, k_s$ is increasing. Furthermore

$$k_1 = |U_1| + 1 \geq 1 + 1 = 2$$

and

$$k_s = \sum_{i=1}^{s} |U_i| + 1 = n - |U_{s+1}| + 1 \leq n - 1 + 1 = n$$

This shows that the set $F = \{k_1, \ldots, k_s\}$ has the properties of an n-factorization schema. We say that F is the factorization schema *induced* by U.

Each factorization $w_1 \cdots w_{s+1}$ of a word w induces a partition or a "factorization" of the vector $\Psi(w) = \Psi(w_1) + \cdots + \Psi(w_{s+1})$ in the monoid $M = \mathbb{N}^n$. As the following example shows the opposite is not true, i.e. a factorization of $\Psi(w)$ does not necessarily correspond to a factorization of w.

Example 4.7 *Let $\Sigma = \{a, b, c\}$ and $w = aabca$. Then*

$$\Psi(w) = (3,1,1) = (2,0,1) + (1,1,0) \tag{60}$$

but $\Psi^{-1}((2,0,1)) = \{aac, aca, caa\}$ does not contain a factor of w.

Note that when we are speaking of factorization of elements in $\mathbb{N}^m$ we are really speaking of sums of elements. These "factorizations" are commutative while the factorizations of words in Σ^* are not.

4.6 Joins of Semi-Commutative Images

In Section 3.3 we showed how to construct a join of two sequences in general groups and in Section 3.4 we applied this to joins of factorizations. In this section we will construct joins between semi-commutative images and investigate how these relate to joins of factorizations. This is the construction used to join the sequences mentioned in the introduction of this paper.

Let $m = |\Sigma|$ and $M = \mathbb{Z}^m$. We consider M as a group under ordinary vector addition. Take $U = (U_1, \ldots, U_s) \in M^s$ and $V = (V_1, \ldots, V_t) \in M^t$ where s and t are two positive integers. U and V are then two sequences in M. We will use the norm

$$|X|_M = \sum_{i=1}^{m} |x_i|. \tag{61}$$

for elements $X = (x_1, \ldots, x_m) \in M$. Assume $n = \sum_{i=1}^{s} |U_i|_M = \sum_{i=1}^{t} |V_i|_M$. Construct $\varphi_{U,V}$ and Z as in Section 3.3 using the norm $|\cdot|_M$. In general $Z_i \in \mathbb{Z}^m$ for $i = 1, \ldots, r$. Now we are mainly interested in sequences that could be interpreted as semi-commutative images. These sequences are those that belong to $\mathbb{N}^{m*}$. It is natural to require that also the join of these sequences should belong to $\mathbb{N}^{m*}$ and not to $\mathbb{Z}^{m*}$. To achieve this we assume that $\varphi_{U,V}(i) \leq \varphi_{U,V}(i+1)$ for $i = 1, \ldots, r-1$. That is, if $\varphi_{U,V}(i) = (x_1, \ldots, x_n)$ and $\varphi_{U,V}(i+1) = (y_1, \ldots, y_n)$ then $x_j \leq y_j$ for $j = 1, \ldots, n$ and $i = 1, \ldots, r-1$. The monotonicity of $\varphi_{U,V}(i)$ assures that the vector Z'_i as defined in (17) is a vector in $\mathbb{N}^m$. That is,

$$Z_i \in \mathbb{N}^m \quad \text{for} \quad i = 1, \ldots r \tag{62}$$

and therefore the sequence $Z \in (\mathbb{N}^m)^*$. We capture this in the following definition.

Definition 4.8 *Let $m = |\Sigma|$ and $M = \mathbb{Z}^m$. Let $U = (U_1, \ldots, U_s) \in M^s$ and $V = (V_1, \ldots, V_t) \in M^t$ where s and t are two positive integers. Assume $\sum_{i=1}^{s} |U_i|_M = \sum_{i=1}^{t} |V_i|_M$. Then we say that U and V are* compatible *if*

$$\varphi_{U,V}(i) \leq \varphi_{U,V}(i+1) \tag{63}$$

for $i = 1, \ldots, r-1$.

By using the above construction we can define the join of two semi-commutative images. Let us say that $U = \Psi_{\overline{F}}(w)$ and $V = \Psi_{\overline{G}}(w)$. The join $\Psi^*_{\overline{F}}(w) \vee \Psi^*_{\overline{G}}(w)$ then is defined to be the sequence Z in the construction above.

Example 4.9 *Let $\Sigma = \{a, b, c\}$ and $w = aabcabcb$. Also let*

$$\begin{aligned} \overline{F}(w) &= (aa, bc, abc, b) \\ \overline{G}(w) &= (aab, ca, b, cb). \end{aligned}$$

These factorizations induce the following semi-commutative images

$$\begin{array}{rcl} U = \Psi^*_{\overline{F}}(w) & = & (\Psi(aa), \Psi(bc), \Psi(abc), \Psi(b)) \\ & = & ((2,0,0),(0,1,1),(1,1,1),(0,1,0)) \\ V = \Psi^*_{\overline{G}}(w) & = & (\Psi(aab), \Psi(ca), \Psi(b), \Psi(cb)) \\ & = & ((2,1,0),(1,0,1),(0,1,0),(0,1,1)). \end{array}$$

The construction yields

$$\begin{array}{ll} U'_1 = (2,0,0) & V'_1 = (2,1,0) \\ U'_2 = U'_1 + (0,1,1) = (2,1,1) & V'_2 = V'_1 + (1,0,0) = (3,1,1) \\ U'_3 = U'_2 + (1,1,1) = (3,2,2) & V'_3 = V'_2 + (0,1,0) = (3,2,1) \\ U'_4 = U'_3 + (0,1,0) = (3,3,2) & V'_4 = V'_3 + (0,1,1) = (3,3,2). \end{array}$$

After ordering the elements U'_i and V'_i with the norm $|U| = \sum |u_i|$ we get

$$\begin{array}{ll} \varphi_{U,V}(1) = U'_1 & \varphi_{U,V}(5) = V'_3 \\ \varphi_{U,V}(2) = V'_1 & \varphi_{U,V}(6) = U'_3 \\ \varphi_{U,V}(3) = U'_2 & \varphi_{U,V}(7) = U'_4 \\ \varphi_{U,V}(4) = V'_2 & \varphi_{U,V}(8) = V'_4. \end{array}$$

We note that $\varphi_{U,V}(i) \leq \varphi_{U,V}(i+1)$ for $i = 1, \ldots, 7$ so U and V are compatible. We continue with the construction and get

$$\begin{array}{rll} Z'_1 = & U'_1 & = (2,0,0) \\ Z'_2 = & -U'_1 + V'_1 = (-2,0,0) + (2,1,0) & = (0,1,0) \\ Z'_3 = & -V'_1 + U'_2 = (-2,-1,0) + (2,1,1) & = (0,0,1) \\ Z'_4 = & -U'_2 + V'_2 = (-2,-1,-1) + (3,1,1) & = (1,0,0) \\ Z'_5 = & -V'_2 + V'_3 = (-3,-1,-1) + (3,2,1) & = (0,1,0) \\ Z'_6 = & -V'_3 + U'_3 = (-3,-2,-1) + (3,2,2) & = (0,0,1) \\ Z'_7 = & -U'_3 + U'_4 = (-3,-2,-2) + (3,3,2) & = (0,1,0) \\ Z'_8 = & -U'_4 + V'_4 = (-3,-3,2) + (3,3,2) & = (0,0,0). \end{array}$$

The unity element in this example is $(0,0,0)$. When the unity elements are removed we finally get $U \vee V = (Z'_1, \ldots, Z'_7)$.

4.7 Joins of Factorizations and Semi-Commutative Images

We are now ready to show the relation between joins of factorizations and joins of semi-commutative images. We start by proving some lemmas.

Lemma 4.10 *Assume $|\Sigma| = m$. Let $M = \mathbb{N}^m$. Let s and t be two positive integers and $U \in M^s$ and $V \in M^t$ be two sequences such that*

$\sum |U_i| = \sum |V_i| = n$. *Let F_U and F_V be the corresponding induced factorization schemata. Then U and V are compatible if and only if*

$$\Psi^{*-1}_{\overline{F_U} \vee \overline{F_V}}(U \vee V) \neq \emptyset. \tag{64}$$

Proof. Let $Z_1 = \varphi(1)$ and $Z_i = \varphi^{-1}(i-1)\varphi(i)$ for $i = 2, \ldots, s$ as in the construction of $U \vee V$. First assume that U and V are compatible. Let us say $\Sigma = \{a_1, \ldots, a_m\}$. Since U and V are compatible we have $Z_i \in \mathbb{N}^m$ where $Z_i = (c_1, \cdots, c_m)$ for some $c_i \in \mathbb{N}$. Then we can construct a word $w_i = a_1^{c_1} \cdots a_m^{c_m}$ and clearly $w_i \in \Psi^{-1}(Z_i)$ for $i = 1, \ldots, s$. Now since at least for one Z_i (e.g. Z_0) we have $c_i \neq 0$ and therefore $w_i \neq \epsilon$. This shows that $\pi(w_1, \ldots, w_s)$ belongs to the set in the left-hand side of equation (64).

Now assume $\Psi^{*-1}_{\overline{F_U} \vee \overline{F_V}}(U \vee V)$ is non-empty so that w belongs to the set. Let $(\overline{F_U} \vee \overline{F_V})(w) = w_1 \cdots w_s$. Then

$$\varphi(i) = \Psi(w_1 \cdots w_i) \leq \Psi(w_1 \cdots w_{i+1}) = \varphi(i+1)$$

Which shows that U and V are compatible. □

Lemma 4.11 *Let $w, u, v \in \Sigma^*$. If $w = uv$ then*

$$\Psi(v) = \Psi(u^{-1}w) = -\Psi(u) + \Psi(w) \tag{65}$$

where u^{-1} is the inverse of u in $\langle \Sigma \rangle$.

Proof. Let $u = u_1 \cdots u_s$ with $u_i \in \Sigma$ for $i = 1, \ldots, s$. Then from (42) we get

$$\begin{aligned} \Psi(v) &= \Psi(u^{-1}uv) \\ &= \Psi(u^{-1}) + \Psi(uv) \\ &= \Psi(u_s^{-1} \cdots u_1^{-1}) + \Psi(w) \\ &= -\Psi(u_s \cdots u_1) + \Psi(w) \\ &= -\Psi(u) + \Psi(w) \end{aligned}$$

Where we used the obvious fact that the commutative image of a word remains the same when the order of the letters is reversed. □

The following theorem shows that the join of two semi-commutative images induced by a factorization is the same as the semi-commutative image of the join of the corresponding factorizations.

Theorem 4.12 *Let $w \in \Sigma^n$ and F and G be two n-factorization schemata and take $w \in \Sigma^n$. Then*

$$\Psi^*_{\overline{F} \vee \overline{G}}(w) = \Psi^*_{\overline{F}}(w) \vee \Psi^*_{\overline{G}}(w). \tag{66}$$

Proof. Let $U = \overline{F}(w) = (u_1, \ldots, u_{s+1})$, $V = \overline{G}(w) = (v_1, \ldots, v_{t+1})$. By Theorem 3.20 we get $(\overline{F} \vee \overline{G})(w) = (z_1, \ldots, z_{p+1})$. By the proof of the same theorem we have $\bar{U}_i = u_1 \cdots u_i$ and $\bar{V}_j = v_1 \cdots v_j$ and $\varphi_{U,V}$ is the ordering function of the elements $\bar{U}_i$ and $\bar{V}_j$ such that

$$z_i = \varphi_{U,V}(i-1)^{-1}\varphi_{U,V}(i) \tag{67}$$

for $i = 1, \ldots, p+1$. We also note that $\varphi_{U,V}(i-1)$ is a prefix of $\varphi_{U,V}(i)$ by Corollary 3.22. By definition we have

$$\Psi^*_{\overline{F}\vee\overline{G}}(w) = (\Psi(z_1), \ldots, \Psi(z_{p+1})). \tag{68}$$

We now turn to the construction of $\Psi^*_{\overline{F}}(w) \vee \Psi^*_{\overline{G}}(w)$. Let $U' = \Psi^*_{\overline{F}}(w) = (\Psi(u_1), \ldots, \Psi(u_{s+1}))$ and $V' = \Psi^*_{\overline{G}}(w) = (\Psi(v_1), \ldots, \Psi(v_{t+1}))$. Let $U'_i = \Psi(u_i)$ for $i = 1, \ldots, s+1$ and $V'_j = \Psi(v_j)$ for $j = 1, \ldots, t+1$ and define $\bar{U}'_i = U'_1 + \cdots + U'_i$ and $\bar{V}'_j = V'_1 + \cdots + V'_j$. Let

$$Z' = (z'_1, \ldots, z'_{p'+1}) = \Psi^*_{\overline{F}}(w) \vee \Psi^*_{\overline{G}}(w) \tag{69}$$

be the join defined in Section 3.3 and $\varphi_{U',V'}$ be the ordering of the elements in $\{\bar{U}'_i\} \cup \{\bar{V}'_j\}$. Then

$$z'_i = -\varphi_{U',V'}(i-1) + \varphi_{U',V'}(i). \tag{70}$$

Now

$$|\bar{U}'_i| = |\Psi(u_1) + \cdots + \Psi(u_i)| = |\Psi(u_1 \cdots u_i)| = |\bar{U}_i|$$

so the ordered elements of $\{\bar{U}'_i\} \cup \{\bar{V}'_j\}$ and $\{\bar{U}_i\} \cup \{\bar{V}_j\}$ are put in a 1-1 correspondence by the following equality

$$\varphi_{U',V'}(i) = \Psi(\varphi_{U,V}(i)). \tag{71}$$

Now, using equations (70) and (71) and the fact that $\varphi_{U,V}(i-1)$ is a prefix of $\varphi_{U,V}(i)$ Lemma 4.11 and equation (67) show that

$$\begin{aligned} z'_i &= -\varphi_{U',V'}(i-1) + \varphi_{U',V'}(i) \\ &= -\Psi(\varphi_{U,V}(i-1)) + \Psi(\varphi_{U,V}(i)) \\ &= \Psi(\varphi_{U,V}(i-1)^{-1}\varphi_{U,V}(i)) \\ &= \Psi(z_i). \end{aligned} \tag{72}$$

Finally this gives us

$$\Psi^*_{\overline{F}}(w) \vee \Psi^*_{\overline{G}}(w) = (\Psi(z_1), \ldots, \Psi(z_{p+1})) = \Psi^*_{\overline{F}\vee\overline{G}}(w). \tag{73}$$

□

Remark 4.13 *In the second equality in (72) one might think that just by altering the order of the terms in the sum one would get* $\Psi(\varphi_{U,V}(i)) - \Psi(\varphi_{U,V}(i-1)) = \Psi(\varphi_{U,V}(i)\varphi_{U,V}(i-1)^{-1})$*, which is not* $\Psi(z_i)$*. However,* $\varphi_{U,V}(i)$ *is not a prefix of* $\varphi_{U,V}(i-1)$ *so Lemma 4.11 does not apply.*

Corollary 4.14 *We have*

$$\Psi^*_{\overline{F}}(w) \vee \Psi^*_{\overline{G}}(w) = \Psi^*(\overline{F}(w) \vee \overline{G}(w)). \tag{74}$$

Proof. Theorem 3.20 gives

$$\begin{aligned} \Psi^*(\overline{F}(w) \vee \overline{G}(w)) &= \Psi^*((\overline{F} \vee \overline{G})(w)) \\ &= \Psi^*_{\overline{F} \vee \overline{G}}(w) \\ &= \Psi^*_{\overline{F}}(w) \vee \Psi^*_{\overline{G}}(w) \end{aligned}$$

□

Theorem 4.15 *Let* $U, V \in (\mathbb{N}^m)^{s+1}$ *and* $\sum |u_i| = \sum |v_i| = n$ *where* $U = (u_1, \ldots, u_{s+1})$ *and* $V = (v_1, \ldots, v_{s+1})$*. Then* U *and* V *are compatible if there exists a word* $w \in \Sigma^n$ *and* $F, G \in \mathcal{F}_n$ *such that* $\Psi^*_{\overline{F}}(w) = U$ *and* $\Psi^*_{\overline{G}}(w) = V$*.*

Proof. Assume $w \in \Sigma^n$ and $F, G \in \mathcal{F}_n$ such that $\Psi^*_{\overline{F}}(w) = U$ and $\Psi^*_{\overline{G}}(w) = V$. Then by Theorem 4.12 we have $U \vee V = \Psi^*_{\overline{F}}(w) \vee \Psi^*_{\overline{F}}(w) = \Psi^*_{\overline{F} \vee \overline{G}}$. But then $w \in \Psi^{*-1}_{\overline{F} \vee \overline{G}}(U \vee V)$ so by Lemma 4.10 we see that U and V are compatible. □

We illustrate the theorem by continuing the previous example.

Example 4.16 (Example 4.9 Continued) *We have*

$$(\overline{F} \vee \overline{G})(w) = (aa, b, c, a, b, c, b) \tag{75}$$

thus

$$\begin{aligned} \Psi^*_{\overline{F} \vee \overline{G}}(w) &= (\,(2,0,0), (0,1,0), (0,0,1), (1,0,0), \\ &\qquad (0,1,0), (0,0,1), (0,1,0)\,) \end{aligned}$$

which is exactly the sequence $\Psi^*_{\overline{F}}(w) \vee \Psi^*_{\overline{G}}(w)$ *constructed in Example 4.9.*

Theorem 3.20 and Theorem 4.12 are illustrated by the following commutative diagram.

$$\begin{tikzcd}
& \Sigma^{**} \arrow[rr, "\Psi^*"] & & \mathbb{N}^{m*} \\
\Sigma^* \arrow[ur, "\overline{F}\vee\overline{G}"] \arrow[dr, "(\overline{F}{,}\overline{G})"'] & & & \\
& \Sigma^{**} \times \Sigma^{**} \arrow[uu, "\vee"] \arrow[rr, "(\Psi^*{,}\Psi^*)"] & & \mathbb{N}^{m*} \times \mathbb{N}^{m*} \arrow[uu, "\vee"]
\end{tikzcd}$$

We have stated several results concerning the relation between joins of factorizations and joins of semi-commutative images. The following theorem describes how these joins behave when taking inverses of semi-commutative images.

Theorem 4.17 *Let Σ be an alphabet with m letters. Let $M = \mathbb{N}^m$. Let s and t be two positive integers and $U \in M^s$ and $V \in M^t$ be two compatible sequences such that $\sum |U_i| = \sum |V_i| = n$. Let F_U and F_V be the factorization schemata induced by U and V. Then*

$$\Psi^{*-1}_{\overline{F_U}\vee\overline{F_V}}(U \vee V) = \Psi^{*-1}_{\overline{F_U}}(U) \cap \Psi^{*-1}_{\overline{F_V}}(V). \tag{76}$$

Proof. By Lemma 4.10 we know that $\Psi^{*-1}(U \vee V)$ is non-empty so we may take $w \in \Psi^{*-1}_{\overline{F_U}\vee\overline{F_V}}(U \vee V)$. Then

$$\Psi^{*}_{\overline{F_U}\vee\overline{F_V}}(w) = U \vee V \tag{77}$$

so

$$\mathcal{S}_{\overline{F_U}\vee\overline{F_V}}(w) = \Psi^{*-1}_{\overline{F_U}\vee\overline{F_V}}(U \vee V). \tag{78}$$

Now $\overline{F_U} \leq \overline{F_U} \vee \overline{F_V}$ so by Theorem 4.5

$$\mathcal{S}_{\overline{F_U}\vee\overline{F_V}}(w) \subset \mathcal{S}_{\overline{F_U}}(w) = \Psi^{*-1}_{\overline{F_U}}(\Psi^{*}_{\overline{F_U}}(w)) = \Psi^{*-1}_{\overline{F_U}}(U) \tag{79}$$

and similarly

$$\mathcal{S}_{\overline{F_U}\vee\overline{F_V}}(w) \subset \Psi^{*-1}_{\overline{F_V}}(V),$$

which shows

$$w \in \Psi^{*-1}_{\overline{F_U}}(U) \cap \Psi^{*-1}_{\overline{F_V}}(V).$$

We have shown

$$\Psi^{*-1}_{\overline{F_U}\vee\overline{F_V}}(U \vee V) \subset \Psi^{*-1}_{\overline{F_U}}(U) \cap \Psi^{*-1}_{\overline{F_V}}(V) \tag{80}$$

Now conversely, equation (80) also shows that the right hand side is non-empty. So we may take $w \in \Psi^{*-1}_{\overline{F_U}}(U) \cap \Psi^{*-1}_{\overline{F_V}}(V)$. Then $\Psi^{*}_{\overline{F_U}}(w) = U$ and $\Psi^{*}_{\overline{F_V}}(w) = V$. By Theorem 4.12

$$U \vee V = \Psi^{*}_{\overline{F_U}}(w) \vee \Psi^{*}_{\overline{F_V}}(w) = \Psi^{*}_{\overline{F_U}\vee\overline{F_V}}(w) \tag{81}$$

which shows that $w \in \Psi^{*-1}_{\overline{F_U}\vee\overline{F_V}}(U \vee V)$. □

Corollary 4.18 *With U and V as in the previous theorem*

$$\Psi_{F_U}^{*-1}(U) \cap \Psi_{F_V}^{*-1}(V) \neq \emptyset. \tag{82}$$

Proof. This follows directly from Lemma 4.10. □

5 Decompositions of Monoids

As stated in the introduction, the goal behind the idea of this paper is to reconstruct a word given some semi-commutative images of the word. In this section we will discuss some properties of the corresponding factorizations which ensures that the reconstruction is unique. The idea is to find a family of sets such that if the word is factorized over this family the join of the corresponding semi-commutative images will be a Σ-factorization of the word. Or more precisely a Σ°-factorization, where Σ° is the set of all words consisting of just one of the letters in Σ.

Lemma 5.1 *Let $X_i \subset \Sigma^*$ for $i = 1, \ldots, n$. Let $w \in \Sigma^*$. Then w have an X_i-factorization x_i for each $i = 1, \ldots, n$ iff $w \in \cap_{i=1}^n X_i^*$.*

Proof. Take $w \in \Sigma^*$ such that w has an X_i-factorization for all $i = 1, \ldots, n$. Then $w = \pi(x_i) \in X_i^*$ for all $i = 1, \ldots, n$. This proves $w \in \cap_{i=1}^n X_i^*$. Take $w \in \cap_{i=1}^n X_i^*$ then since $w \in X_i^*$ for each $i = 1, \ldots, n$ there are words $x_j^{(i)} \in X_i$ for $j = 1, \ldots, s_i$ such that $w = x_1^{(i)} \cdots x_{s_i}^{(i)}$ and this gives an X_i-factorization of w. □

Definition 5.2 (Decomposition) *Let $X_i, Y \subset \Sigma^*$ for $i = 1, ..n$. Assume that for all words $w \in \cap_{i=1}^n X_i^*$ and any family $\{x_i\}_{i=1}^n$ of factorizations of w where x_i is an X_i-factorization, the join $\vee_{i=1}^n x_i$ is a Y-factorization of w. Then we call $\{X_i\}_{i=1}^n$ a Y-*decomposition *of Σ^*.*

We begin with some general properties of decompositions.

Theorem 5.3 *Let $X_i, Y \subset \Sigma^*$ for $i = 1, \ldots, n$. If $\{X_i\}_{i=1}^n$ is a Y-decomposition of Σ^* then*

$$\bigcap_{i=1}^n X_i^* \subset Y^* \tag{83}$$

Proof. By Lemma 5.1 the intersection on the left is the set of all $w \in \Sigma^*$ such that w has a X_i-factorization x_i for each $i = 1, \ldots, n$. Take one such w. By the definition of decomposition the join $y = \vee_{i=1}^n x_i$ is a Y-factorization so clearly $w = \pi(y)$ belongs to Y^*. □

For an alphabet Σ we define

$$\Sigma^\circ = \bigcup_{a\in\Sigma} \{a\}^* \tag{84}$$

Clearly every word $w \in \Sigma^*$ has a unique minimal Σ°-factorization, say $(w_1, \ldots, w_k)$, where each w_i is a power of a letter in Σ and w_i and w_{i+1} are not powers of the same letter.

Lemma 5.4 *Let $\Sigma = \{a, b, c, d\}$. Define $X_1 = \{a, b\}$, $X_2 = \{c, d\}$, $Y_1 = \{a, c\}$ and $Y_2 = \{b, d\}$. Let $X = X_1^* \cup X_2^*$ and $Y = Y_1^* \cup Y_2^*$. If $x \in X$ is a subword of $y \in Y$ then $x \in \Sigma^\circ$.*

Proof. That x is a subword of y means that there exist u and v such that $y = uxv$. Assume $y \in Y_1^*$. Then since Y_1 is generated by letters we have $u, x, v \in Y_1^*$. This means that

$$\begin{aligned} x \in X \cap Y_1^* &= (X_1^* \cup X_2^*) \cap Y_1^* \\ &= (X_1^* \cap Y_1^*) \cup (X_2^* \cap Y_1^*) \\ &= \{a\}^* \cup \{c\}^* \end{aligned}$$

which shows $x \in \Sigma^\circ$. Similarly the assumption $y \in Y_2$ shows $x \in \{b\}^* \cup \{d\}^* \subset \Sigma^\circ$. □

Theorem 5.5 *Let $\Sigma = \{a, b, c, d\}$. Define $X_1 = \{a, b\}$, $X_2 = \{c, d\}$, $Y_1 = \{a, c\}$ and $Y_2 = \{b, d\}$. Let $X = X_1^* \cup X_2^*$ and $Y = Y_1^* \cup Y_2^*$. Then $\{X, Y\}$ is a Σ°-decomposition of Σ^*.*

Proof. Assume $w \in \Sigma^*$. Let $u = (u_1, \ldots, u_s)$ and $v = (v_1, \ldots, v_t)$ be two arbitrary X and Y-factorizations respectively of w.

Let $z = (z_1, \ldots, z_k) = u \vee v$. Then $z_1 = \varphi_{u,v}(1)$ and $z_i = \varphi_{u,v}(i-1)^{-1}\varphi_{u,v}(i)$ for $i = 2, \ldots, k$. Without loss of generality we may assume $z_1 = \varphi_{u,v}(1) = u_1$. Thus $z_1 \in X = \{a, b\}^* \cup \{c, d\}^*$. We may assume $z_1 = \{a, b\}^*$. If $z_1 \in \{a\}^*$ or $z_1 \in \{b\}^*$ then $z_1 \in \Sigma^\circ$. Otherwise, without loss of generality, we may assume $z_1 = a^c bx$, for some integer $c \geq 1$ and $x \in \{a, b\}^*$. But then we must have $v_1 = a^c$ which contradicts $|u_1| = |\varphi_{u,v}(1)| < |v_j|$ for all $j = 1, \ldots, t$. We have shown $z_1 \in \Sigma^\circ$. Assume that $z_i \in \Sigma^\circ$ for $i = 1, \ldots, j-1$. We will show that $z_j \in \Sigma^\circ$. We have $z_j = \varphi_{u,v}(j-1)^{-1}\varphi_{u,v}(j)$. If $z_j \in \Sigma^\circ$ we are done. Otherwise z is a product of two letters. It does not matter which two and which order so we may assume $z_j \in \{a, b\}^*$ and that $z_j = a^{c'}bx'$ for some $c' \geq 1$ and $x' \in \{a, b\}^*$. This would mean that $\varphi_{u,v}(j) = u_1 \cdots u_{k'}$ for some k'. But then $\varphi_{u,v}(j-1)a^c = v_1 \cdots v_{k''}$ for some k''. Thus $|\varphi_{u,v}(j-1)| < |v_{k''}| < |\varphi_{u,v}(j)|$ which is a contraction. Therefore $z_j \in \Sigma^\circ$. □

The following theorem shows that if we are given two semi-commutative images induced by factorizations over a Σ°-decomposition, then their inverse sets only have one word in common. We start by proving some lemmas.

Lemma 5.6 *Let $\overline{F}(w)$ be a Σ°-factorization of w. Then $\Psi^*_{\overline{F}}(w)$ is a sequence of base vectors in $\mathbb{N}^m$ each scaled by a non-negative integer.*

Proof. We have
$$\overline{F}(w) = (a_{i_1}^{c_1}, \ldots, a_{i_n}^{c_n})$$
where $a_{i_j} \in \Sigma$ and $c_j \in \mathbb{N}$ for $j = 1, \ldots, n$. Thus
$$\Psi^*_{\overline{F}}(w) = (c_1\Psi(a_{i_1}), \ldots, c_n\Psi(a_{i_n})).$$
□

Theorem 5.7 *If $\overline{F}(w)$ is a Σ°-factorization of w then*
$$|\Psi^{*-1}_{\overline{F}}(w)| = 1.$$

Proof. This follows directly from Lemma 5.6 and equation (47). □

Theorem 5.8 *Let $w \in \Sigma^n$ and $\overline{F}(w)$ and $\overline{G}(w)$ be two X and Y factorizations respectively. Let $U = \Psi^*_{\overline{F}}(w)$ and $V = \Psi^*_{\overline{G}}(w)$. If $\{X, Y\}$ is a Σ°-decomposition then*
$$\Psi^{*-1}_{\overline{F}\vee\overline{G}}(U \vee V) = \Psi^{*-1}_{\overline{F}}(U) \cap \Psi^{*-1}_{\overline{G}}(V) = \{w\}. \tag{85}$$

Proof. The first equality is just Theorem 4.17 and clearly
$$w \in \Psi^{*-1}_{\overline{F}}(U) \cap \Psi^{*-1}_{\overline{G}}(V).$$
Since $\{X, Y\}$ is a Σ°-decomposition, $(\overline{F} \vee \overline{G})(w')$ is a Σ°-factorization of w so Theorem 5.8 implies that w is the only element in the set. □

6 Conclusion and Further Research

In this paper we have shown that given some semi-commutative images of factorizations of a word, where the factorizations are taken over a Σ°-decomposition, we are able to reconstruct the word by considering the inverse image of the join of the semi-commutative images.

Let us now return to the biological problem described in the introduction. We then see that the above result applies to the situation where the scanner delivers precise and error-free measures. Below we will describe some first steps in the direction of dealing with the situation where the measurements contain errors. The signal vector, or the semi-commutative image, of the word thus gives a more or less accurate indication of the number of each letter in the word. Our interest lies in the set of possible words that could result in this signal vector. To study this set we introduce the concepts of commutative and semi-commutative d-closures. This is the set of words that are “close” to another word according to some notion of distance. Some possible distance functions are suggested. Also some natural generalizations of $\mathcal{X}$-factorizations and decompositions are given. In Appendix A an alternative way of dealing with the problem of inexact measurements is described.

6.1 Parikh Distances

We begin with some definitions of distance between words. In the following we will make use of the following norm on $\mathbb{Z}^m$. For $u = (u_1, \ldots, u_m) \in \mathbb{Z}^m$ we define

$$|u| = \sum_{i=1}^{m} |u_i|$$

Definition 6.1 (Parikh distance) *For two words $w, w' \in \Sigma^*$ we define the* Parikh distance

$$\mathcal{P}(w, w') = |\Psi(w) - \Psi(w')|$$

We generalize the above definition to a function where differences between different letters could be weighted.

Definition 6.2 (Weighted Parikh distance) *For $|\Sigma| = m$ and $v \in \mathbb{R}^m$ we define*

$$\mathcal{P}_v(w, w') = |v \odot (\Psi(w) - \Psi(w'))|$$

where $\odot$ represent element-wise multiplication.

The following distance function weights both the letters and the difference in length between two words.

Definition 6.3 (Length and Parikh distance) *For $d, d_l \in \mathbb{Z}$, $v \in \mathbb{R}^m$ and $v_l \in \mathbb{R}$ let*

$$\mathcal{P}^{d,d_l}_{v,v_l}(w, w') = |v \odot (\Psi(w) - \Psi(w')|^d + (v_l \odot |\,|\Psi(w)| - |\Psi(w')|\,|)^{d_l}$$

6.2 Commutative d-closures

We will make use of the previously defined distance functions to give generalized definitions of commutative and semi-commutative closures.

Definition 6.4 (Commutative d-closure) *Let $\mathcal{P}$ be any distance function defined on Σ^*. For $w \in \Sigma^*$ and $d \in \mathbb{R}$ with $d \geq 0$ we call*

$$\mathcal{C}_d(w) = \{w' \in \Sigma^* | \mathcal{P}(w, w') \leq d\}$$

the commutative d-closure *of w with respect to $\mathcal{P}$.*

Definition 6.5 (Semi-commutative d-closure) *Let $w \in \Sigma^n$ and $\overline{F} \in \mathcal{F}_n$ so that $\overline{F}(w) = w_1 \cdots w_s$. Let $\mathcal{P}$ be a distance function defined on Σ^*. For $d \in \mathbb{R}$ with $d \geq 0$ we call*

$$\mathcal{S}_{\overline{F},d}(w) = \{w' \in \Sigma^n | w' = w'_1 \cdots w'_s, \mathcal{P}(w_i, w'_i) \leq d\}$$

the semi-commutative d-closure *of w with respect to $\mathcal{P}$.*

6.3 $\mathcal{X}_d$-Factorizations and Y_d-Decompositions

It is also possible to generalize $\mathcal{X}$-factorizations and Y-decompositions.

Definition 6.6 ($\mathcal{X}_d$-factorization) *Let $\mathcal{P}$ be a distance function on Σ^*. Let $\mathcal{X} = \{X_i\}_{i=1}^s$ for $i = 1, \ldots, s$ where $X_i \subset \Sigma^*$. Take $w \in \Sigma^*$ with a factorization $w = w_1 \cdots w_s$. If there is $w'_i \in X_i$ and $d \in \mathbb{R}$ with $d \geq 0$ such that $\mathcal{P}(w_i, w'_i) \leq d$ for $i = 1, \ldots, s$, then we call $w_1 \cdots w_s$ a* $\mathcal{X}_d$-factorization *of w with respect to $\mathcal{P}$.*

Definition 6.7 (Y_d-Decompositions) *Let $X_i, Y \subset \Sigma^*$ for $i = 1, ..n$. Assume that for all words $w \in \cap_{i=1}^n X_i^*$ and any family $\{x_i\}_{i=1}^n$ of factorizations of w where x_i is an X_{i,d_i}-factorization for some fixed $d_i \in \mathbb{R}$, $d_i \geq 0$. Further assume that the join $\vee_{i=1}^n x_i$ is a Y_d-factorization of w for some fixed $d \in \mathbb{R}$, $d \geq 0$. Then we call $\{X_i\}_{i=1}^n$ a* Y_d-decomposition *of Σ^*.*

A Fuzzy Languages

A.1 Introduction

In this appendix we will give an alternative way of modeling the problem given in the introduction of this paper.

We will introduce a model for string representation of polynucleotide sequences such as DNA and RNA where the information about the exact sequences are more or less certain. The theory of fuzzy sets will be used to model this uncertainty of the sequence.

The situation in the reconstruction problem is that we have several partial descriptions of a DNA-sequence. By this we mean that each partial description on its own is not enough to give an exact description of the sequence. We will use the concept of fuzzy languages to model this uncertainty about the information. The membership of each word in the language then represents how likely it is that that word is indeed the sequence we want.

We assume that the partial information is represented by sequences of signal vectors as described in the introduction. We will show how to generate a fuzzy language from such a signal vector. We will also show that by taking the intersection of several such fuzzy languages the fuzziness of the resulting fuzzy language decreases.

A.2 Fuzzy Sets

The notion of a fuzzy subset of a set was introduced by Zadeh in [Zad65].

Definition A.1 *A fuzzy subset A of a set X is a function*

$$\mu_A : X \to [0,1] \tag{86}$$

where $[0,1]$ *denotes the closed interval of real numbers between* 0 *and* 1.

For a given $x \in X$ the function μ_A of a fuzzy subset A can be thought of as a measure of the degree of membership of x in A. μ_A should be considered a generalization of the *characteristic functions* used to define ordinary subsets of X. One would interpret $\mu_A(x) = 1$ as that x belongs to the set and $\mu_A(x) = 0$ would mean that x is not a member of the set.

Ordinary subsets are often referred to as *crisp* in the fuzzy set theory. For the basic theory of fuzzy sets see for example the original article of Zadeh [Zad65] where the concept of fuzzy sets first was introduced in the literature.

We continue with some basic definitions.

Definition A.2 *Let A be a fuzzy subset of X.*

1. *Let $\alpha \in [0,1]$. Define $\mu_{A_\alpha} = \{x \in X | \mu_A(x) \geq \alpha\}$. We call the fuzzy subset A_α described by μ_{A_α} for a α-*cut*.*
2. *We define the* support *of A as the set,*

$$\text{supp}(A) = \{x \in X | \mu_A(x) > 0\}.$$

A subset A with a finite support is conveniently denoted as

$$\{(x_i, \mu_A(x_i)) | x_i \in X, \mu_A(x_i) > 0\}.$$

An example of this is

$$\{(-2, 1/3), (3, 2/3)\}$$

Definition A.3 *Let A and B be two fuzzy subsets of X. We define*

1. *the* complement *$\bar{A}$ of A by*

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x).$$

2. *the* intersection *$A \cap B$ of A and B by*

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)).$$

3. *the* union *$A \cup B$ of A and B by*

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)).$$

for all $x \in X$.

We will sometimes use $\wedge$ to denote the min function and $\vee$ to denote the max function. With these symbols we could write

$$\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x) \tag{87}$$

$$\mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x) \tag{88}$$

The notion of union and intersection could easily be extended to a family of fuzzy subsets of X. Let I be an index set and $\{A_i\}_{i \in I}$ be a family of fuzzy subsets of X. For $x \in X$ we then write $\mu_{\cap_{i \in I} A_i}(x) = \wedge_{i \in I} \mu_{A_i}(x)$ and $\mu_{\cup_{i \in I} A_i}(x) = \vee_{i \in I} \mu_{A_i}(x)$.

A.3 Measures of Fuzziness

A.3.1 Distance between fuzzy sets

From now on we let X denote a finite set. Consider two fuzzy subsets $F_1, F_2 \subset X$. We want a concept of *distance* between F_1 and F_2. The distance can be defined in several ways. It is noteworthy that once we have a distance function defined for two *fuzzy* sets we immediately have a distance defined for a crisp and a fuzzy set. This follows from the fact that a crisp set can be regarded as a fuzzy set. We also want to determine the crisp set M which is closest to a given fuzzy set F. One common definition is the following

Definition A.4 (Generalized Hamming Distance)
Let F_1 and F_2 be two fuzzy subsets of X. The distance between F_1 and F_2 is defined as

$$d(F_1, F_2) = \sum_{x \in X} |\mu_{F_1}(x) - \mu_{F_2}(x)| \tag{89}$$

The above distance is often called the *Hamming distance* between F_1 and F_2 since it is the natural generalization of Hamming distance in the common set theory. The Hamming distance defined as above fulfills the general properties of a distance function. Another distance function is the following.

Definition A.5 (Generalized Euclidean distance)

$$e(F_1, F_2) = \sqrt{\sum_{x \in X} (\mu_{F_1}(x) - \mu_{F_2}(x))^2} \tag{90}$$

From the definition it is clear that

$$0 \leq e(F_1, F_2) \leq \sqrt{|X|} \tag{91}$$

where $|X|$ denotes the cardinality of X. Below we give the definition of the corresponding relative distances.

Definition A.6 (Generalized relative Hamming distance)

$$\delta(F_1, F_2) = \frac{d(F_1, F_2)}{|X|} = \frac{1}{|X|} \sum_{x \in X} |\mu_{F_1}(x) - \mu_{F_2}(x)| \tag{92}$$

Definition A.7 (Generalized relative Euclidean distance)

$$\epsilon(F_1, F_2) = \frac{e(F_1, F_2)}{\sqrt{|X|}} = \sqrt{\frac{1}{|X|} \sum_{x \in X} |\mu_{F_1}(x) - \mu_{F_2}(x)|} \tag{93}$$

For further details about distance functions see [Kau75]. We here restate some other definitions and facts presented in [Kau75].

Theorem A.8 (Crisp subset nearest to a fuzzy subset) *The crisp subset M nearest to a fuzzy subset F is given by the membership function*

$$\begin{array}{lllllll} \mu_M(x_i) & = & 0 & \textit{if} & \mu_F(x_i) & < & 0.5, \\ & = & 1 & \textit{if} & \mu_F(x_i) & > & 0.5, \\ & = & 0 \textit{ or } 1 & \textit{if} & \mu_F(x_i) & = & 0.5 \end{array}$$

We follow [Kau75] and define $\mu_M(x_i) = 0$ when $\mu_F(x_i) = 0.5$. One easily sees that

$$|\mu_M(x_i) - \mu_F(x_i)| = \mu_{F \cap \bar{F}}(x_i) \qquad (94)$$

A.3.2 Index of fuzziness

Following [Kau75] we define two measures of the fuzziness of a fuzzy set. The first is defined with respect to the generalized relative Hamming distance, and the other with respect to the relative euclidean distance.

Definition A.9 (Linear Index of fuzziness) *For a fuzzy set F we define the linear index of fuzziness of F as the number*

$$\nu(F) = \frac{2}{|X|} \cdot d(F, M) \qquad (95)$$

where M is the crisp set given in Theorem A.8.

Definition A.10 (Quadratic Index of fuzziness) *For a fuzzy set F we define the quadratic index of fuzziness of F as the number*

$$\eta(F) = \frac{2}{\sqrt{|X|}} \cdot e(F, M) \qquad (96)$$

where M is the crisp set given in Theorem A.8.

The number 2 in the numerator in the above definitions is chosen so that the following inequalities are valid

$$0 \leq \nu(F) \leq 1 \quad \text{and} \quad 0 \leq \eta(F) \leq 1. \qquad (97)$$

Another way of measuring fuzziness is by use of entropy.

Definition A.11 (Entropy as Index of Fuzziness) *Let $F \subset X$ be a fuzzy subset. For each $x \in X$ let*

$$\pi_F(x) = \frac{\mu_F(x)}{\sum_{x' \in X} \mu_F(x')}. \qquad (98)$$

The entropy of F is defined as

$$H(F) = -\frac{1}{\ln |X|} \sum_{x \in X} \pi_F(x) \cdot \ln \pi_F(x). \tag{99}$$

A.4 Fuzzy Languages

A subset $L \subset \Sigma^*$ is often called a *language* in Σ^*. We generalize this to the notion of a *fuzzy language*.

Definition A.12 (Fuzzy Language) *A* fuzzy language *is a fuzzy subset of Σ^*.*

We continue with some basic definitions.

Definition A.13 *Let A and B be two fuzzy languages of Σ^*. We define*

1. *the* intersection *of A and B as the ordinary intersection of fuzzy subsets as defined in A.3.*
2. *the* concatenation *of A and B denoted by AB as*

$$\mu_{AB}(x) = \vee\{\mu_A(u) \wedge \mu_B(v) | x = uv, u, v \in \Sigma^*\}$$

Later we will have to denote the concatenation of a sequence of fuzzy subsets. For the sequence F_i, $i = 1, \dots k$, we will use the following notation

$$\prod_{i=1,\dots,k} F_i = F_1 F_2 \cdots F_k$$

Definition A.14 (Fuzzy Integers) *We define a* fuzzy integer *z as a function*

$$\mu_z : \mathbb{Z} \to [0, 1]. \tag{100}$$

A fuzzy integer is thereby a fuzzy subset of $\mathbb{Z}$.

Example A.15 *For each integer $a \in \mathbb{Z}$ we define a function*

$$\mu_{\tilde{a}}(i) = \begin{cases} 1/3, & \textit{if } |i - a| = 2, \\ 2/3, & \textit{if } |i - a| = 1, \\ 1, & \textit{if } i = a, \\ 0, & \textit{otherwise} \end{cases} \tag{101}$$

for all $i \in \mathbb{Z}$. $\tilde{a}$ is then a fuzzy integer. $\tilde{a}$ should be considered a fuzzy version or a fuzzification *of a. We let $\mathcal{Z} = \{\tilde{a} | a \in \mathbb{Z}\}$ be the corresponding fuzzification of $\mathbb{Z}$. In practice we will write expressions like $\tilde{2}$ when we mean the fuzzification of the integer 2.*

Similarly we define $\mathcal{N} = \{\tilde{a} | a \in \mathbb{N}\}$ where $\mathbb{N}$ denotes the set of all non-negative integers. The elements of $\mathcal{N}$ are considered to be the functions of $\mathcal{Z}$ restricted to $\mathbb{N}$. This gives us e.g.

$$\tilde{0} = \{(-2, 1/3), (-1, 2/3), (0, 1), (1, 2/3), (2, 1/3)\} \tag{102}$$

for $\tilde{0} \in \mathcal{Z}$ but

$$\tilde{0} = \{(0, 1), (1, 2/3), (2, 1/3)\} \tag{103}$$

for $\tilde{0} \in \mathcal{N}$.

A.5 The Shuffle Operator

To continue we will need the following piece of machinery. The definition is due to [Lot97]

Definition A.16 (Shuffle) *The shuffle of two words $f, g \in \Sigma^*$ is the subset of Σ^*, denoted by $f \circ g$ and defined by:*

$$\begin{aligned} f \circ g &= \{h | h = f_1 g_1 f_2 g_2 \cdots f_n g_n, n \geq 0, \\ &\quad f_i, g_i \in \Sigma^*, f = f_1 f_2 \cdots f_n, g = g_1 g_2 \cdots g_n\}. \end{aligned}$$

The shuffle of two subsets F and G of Σ^* is denoted $F \circ G$ and is defined as

$$F \circ G = \bigcup_{f \in F, g \in G} f \circ g$$

We give an easy example of a shuffle.

Example A.17 *The shuffle $aa \circ bb$ is the set*

$$aa \circ bb = \{aabb, abab, baab, abba, baba, bbaa\}$$

One can verify that the shuffle is a commutative and associative operation on the power set $\mathcal{P}(\Sigma^*)$, see for instance [Lot97].

We now proceed and give our fuzzy version of the shuffle.

Definition A.18 (Fuzzy Shuffle) *The shuffle of two fuzzy languages F and G is the fuzzy subset of Σ^* denoted by $F \ominus G$ and defined by:*

$$\mu_{F \ominus G}(h) = \vee\{\mu_F(f) \wedge \mu_G(g) | h = f \circ g, f \in F, g \in G\}.$$

Example A.19 *Let $F = \{(a, 0.5), (\epsilon, 1)\}$ and $G = \{(b, 1), (ab, 1)\}$. Then*

$$F \ominus G = \{(ab, 1), (ba, 0.5), (aab, 0.5), (aba, 0.5), (b, 1)\}$$

A.6 DNA-Sequences as Fuzzy Languages

We are now ready to give an alternative way of modeling the problem given in the introduction to this paper. This way, however, deals with signal vectors that contain errors.

Given a partial description of a DNA-sequence there could be several different sequences which are candidates for the "real" sequence. However, some of the sequences is probably more likely to be the real sequence than others. What we will do is to represent the set of all possible words with a fuzzy language. The membership function of each word then tells how likely it is that the word is the "real" word.

Assume that we have two fuzzy languages S_1 and S_2 each representing partial information about the same word w. By considering the intersection $S_1 \cap S_2$ we will have a new fuzzy language which should represent more precise information about w than S_1 and S_2 does separately. Our goal is to show that the amount of "fuzziness" decreases by taking the intersection and thus that we have constructed a representation of the word with less fuzziness.

Let us first consider how to construct the fuzzy languages from the the partial information. As described in the paper the partial information is given through signal vectors, i.e. vectors with non-negative real values. In the case of DNA-sequences the vectors are 4-dimensional, but for generality we will here discuss vectors in $\mathbb{R}^m$ for any given m.

In the paper we have used the commutative image of a word as a representation of our incomplete knowledge of the word. The commutative image was obtained by the Parikh mapping. We now want to define a fuzzy version of this Parikh mapping. This would give us a fuzzy commutative image. The signal vector could be regarded as the image of a function

$$\sigma : \Sigma^* \to \mathbb{R}^m.$$

The signal vector in it self have real valued positive elements. The corresponding number of letters in the word measured is however always an integer. Our goal is to translate the real valued signal vector into a vector of fuzzy integers. We denote the set of fuzzy integers as described in Example A.15 by $\mathcal{N}$. We assume that we have some map

$$\Phi : \mathbb{R}^m \to \mathcal{N}^m$$

that maps signal vectors into vectors of fuzzy integers. The fuzzy Parikh mapping $\tilde{\Psi}$ would then be defined as

$$\begin{array}{rcl} \tilde{\Psi} : \Sigma^* & \to & \mathcal{N}^m \\ \tilde{\Psi}(w) & = & (\Phi(\sigma(w)). \end{array}$$

It does not matter for us exactly how Φ and σ are defined. The definition of σ depends on which model we choose for the measurement process. The definition of Φ then depends on how we wish to transform the real measurement values into fuzzy integers. Also the choice of how to define the fuzzy integers reflects how we model errors.

We now want to find a way to map vectors in $\mathcal{N}^m$ back to fuzzy languages in Σ^*. To keep it simple, from here on we let $\mathcal{N}$ denote the fuzzy integers defined by

$$\mu_{\tilde{a}}(i) = \begin{cases} 0.6, & \text{if } i = a-1, \\ 1, & \text{if } i = a, \\ 0.4, & \text{if } i = a+1, \\ 0, & \text{otherwise} \end{cases} \tag{104}$$

for all $i, a \in \mathbb{N}$.

We define the following map which could be interpreted as a "multiple fuzzy concatenation". It is a map between the fuzzy integers $\mathcal{N}$ and fuzzy languages in Σ^*. For a fuzzy integer $\tilde{a} \in \mathcal{N}$, $a \geq 1$, we define

$$\Gamma_{\tilde{a}}(w) = \{(w^{a-1}, 0.6), (w^a, 1), (w^{a+1}, 0.4)\}$$

where $w^0 = \epsilon$ and in particular

$$\Gamma_{\tilde{0}}(w) = \{(\epsilon, 1), (w, 0.4)\}.$$

Example A.20 *We have*

$$\Gamma_{\tilde{2}}(ab) = \{(ab, 0.6), (abab, 1), (ababab, 0.4)\}$$

Using the map Γ and the fuzzy shuffle we are now ready to define a mapping from $\mathcal{N}^m$ to fuzzy languages.

We assume that $\Sigma = \{a_1 < \ldots < a_m\}$ is an ordered alphabet. Let $\tilde{U} = (\tilde{u}_1, \ldots, \tilde{u}_m) \in \mathcal{N}^m$. We then define

$$\Gamma_{\ominus,\Sigma}(U) = \Gamma_{\tilde{u}_1}(a_1) \ominus \cdots \ominus \Gamma_{\tilde{u}_m}(a_m)$$

Example A.21 *Let $\Sigma = \{a < b\}$,$U = (\tilde{2}, \tilde{1})$. We then get*

$$\begin{aligned} \Gamma_{\ominus,\Sigma}(U) &= \Gamma_{\tilde{2}}(a) \ominus \Gamma_{\tilde{1}}(b) \\ &= \{(a, 0.6), (aa, 1), (aaa, 0.4)\} \ominus \{(\epsilon, 0.6), (b, 1), (bb, 0.4)\} \\ &= \{(a, 0.6), (ab, 0.6), (abb, 0.4), (bab, 0.4), (bba, 0.4), \\ &\quad (aa, 0.6), (aab, 1), (aba, 1), (baa, 1)\} \cup (aaa \circ bb, 0.4) \end{aligned}$$

where we by $(aaa \circ bb, 0.4)$ mean that $\mu_{aaa \circ bb}(w) = 0.4$ for all $w \in aaa \circ bb$.

A.7 Conclusion and Further Research

The machinery developed in this appendix could be used to reformulate the reconstruction problem discussed in this paper. Let us say that the analysis process σ applied to a word w gives us two sequences of signal vectors $s_1, s_2, \ldots, s_k$ and $t_1, t_2, \ldots t'_k$ in $\mathbb{R}^m$. Our map Φ transforms these vectors into sequences of vectors in $\mathcal{N}^m$. Say for instance that $\Phi(s_i) = u_i$ and $\Phi(t_j) = v_j$ where $u_i, v_j \in \mathcal{N}^m$ for $i = 1, \ldots, k$ and $j = 1, \ldots, k'$. [3] By using shuffle and concatenation, each of these sequences gives us a fuzzy language in Σ^*:

$$S_1 = \prod_{i=1\ldots,k} \Gamma_{\ominus,\Sigma}(u_i)$$

$$S_2 = \prod_{j=1\ldots,k'} \Gamma_{\ominus,\Sigma}(v_i)$$

Now what really interests us is the intersection $S = S_1 \cap S_2$ which is a new fuzzy language. Some questions that we want to answer are the following: The index of fuzziness of S should be less than for S_1 and S_2 and thus closer to its nearest crisp set. How does the closest crisp set to S, say M, differ from the crisp sets closest to S_1 and S_2, say M_1 and M_2? Could it be that all three crisp sets are unequal? And finally, the goal is to find a process so that for the resulting intersection set it holds that $\mu_S(w) = 1$ for the "real" sequence word w and $\mu_S(w') = 0$ for all other words $w' \neq w$.

B Completion of Parikh-matrices

In this appendix we will show yet another way of dealing with the reconstruction problem. This approach makes use of *Parikh matrices* and assumes that the signal vectors contain exact measurements.

The notion of Parikh matrices was introduced in [MSSY00] and further extended in [MSSY00]. A Parikh matrix is a generalization of the Parikh mapping. It is a matrix that contains somewhat more information of the structure of a word than the simple Parikh mapping.

A Parikh matrix mapping is defined through a so called *triangle matrix*. A triangle matrix is a square matrix $P = (p_{i,j})_{1 \leq i,j \leq m}$, such that $p_{i,j}$ is a nonnegative integer for all $1 \leq i, j \leq m$ while $p_{i,j} = 0$ for all $1 \leq j < i \leq m$ and $p_{i,i} = 1$ for all $1 \leq i \leq m$.

[3] One could develop an analogy of the notation in the main part of the paper and introduce the map $\tilde{\Psi}^*_{\overline{F}}$ so that $\tilde{\Psi}^*_{\overline{F}}(w) = (u_1, \ldots, u_k)$ and $\tilde{\Psi}^*_{\overline{G}}(w) = (v_1, \ldots, u_k)$ for some suitable factorization maps $\overline{F}$ and $\overline{G}$.

The set of all triangle matrices is denoted by $\mathcal{M}$. The set of all triangle matrices of dimension $m \geq 1$ is denoted by $\mathcal{M}_m$. Clearly $(\mathcal{M}_m, \cdot, I_m)$, where $\cdot$ represents matrix multiplication and I_m is the unit matrix, is a monoid. The following definition is due to [MSSY00].

Definition B.1 (Parikh Matrix Mapping) *Let $\Sigma = \{a_1 < \cdots < a_m\}$ be an ordered alphabet. The* Parikh matrix mapping*, denoted as $\Psi_{\Sigma,m}$, is the monoid morphism*

$$\Psi_{\Sigma,m} : (\Sigma^*, \cdot, \epsilon) \to (\mathcal{M}_{m+1}, \cdot, I_{m+1}) \tag{105}$$

defined on each letter a_q, $1 \leq q \leq m$ by the following condition: If $\Psi_{\Sigma,m}(a_q) = (p_{i,j})_{1\leq,i,j\leq m+1}$, then for each $1 \leq i \leq (m+1)$, $p_{i,i} = 1$,$p_{q,q+1} = 1$ all other elements of the matrix $\Psi_{\Sigma,m}(a_q)$ being 0.

Note that the matrix mapping is defined on the letters in Σ. The map is then extended to words by the morphism property. In the ordered alphabet $\Sigma = \{a_1 < a_2 < \cdots < a_m\}$ we denote by $a_{i,j}$ the word $a_i a_{i+1} \cdots a_j$, where $1 \leq i \leq j \leq m$. We will use the notation $|w|_{\text{scatt}-f}$ to denote the number of scattered occurrences of the word f in w. The word f occurs as a scatter subword in w if there is a word g such that $w = f \circ g$ where $\circ$ denotes the shuffle operator. We restate Theorem 3.1 of [MSSY00] which will give us the basic property of Parikh matrices.

Theorem B.2 *Let $\Sigma = \{a_1, a_2, \ldots, a_m\}$ be an ordered alphabet and assume that $w \in \Sigma^*$. The matrix $\Psi_{\Sigma,m}(w) = (p_{i,j})_{1\leq i,j\leq(m+1)}$, has the following properties:*

1. *$p_{i,j} = 0$, for all $1 \leq j < i \leq (m+1)$,*
2. *$p_{i,i} = 1$, for all $1 \leq i \leq (m+1)$,*
3. *$p_{i,j+1} = |w|_{\text{scatt}-a_{i,j}}$, for all $1 \leq i \leq j \leq k$.*

Now let us turn to our reconstruction problem. Let $w \in \Sigma$ be a word unknown to us. Let us say that we only know the Parikh-vectors of the factors for some factorizations of w. Our problem is to find a convenient way of expressing the set of possible w' that could be w. In the language of this paper it would be to determine the join of the semi-commutative closures of w induced by the factorizations.

The approach taken here is to express the information about w with the help of incomplete Parikh-matrices.

In the following example we consider the word $w = abbaca \in \Sigma^*$ over the alphabet $\Sigma = \{a, b, c\}$.

We now consider different factorizations of w and investigate the corresponding Parikh-matrices of these factorizations.

Now, we do not know w but let us say that we know that it has a factorization $w = w_1w_2$ where w_1 and w_2 still are unknown to us but we know the corresponding Parikh-vectors $\Psi(w_1) = (1,2,0)$ and $\Psi(w_2) = (2,0,1)$. We can then construct the following *incomplete* Parikh-matrices

$$\begin{pmatrix} 1 & 1 & x_1 & x_3 \\ 0 & 1 & 2 & x_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 2 & y_1 & y_3 \\ 0 & 1 & 0 & y_2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 3 & z_1 & z_3 \\ 0 & 1 & 2 & z_2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The above matrix equation gives us the following equations

$$\begin{aligned} z_1 &= y_1 + x_1 \\ z_2 &= y_2 + 2 + x_2 \\ z_3 &= y_3 + y_2 + x_1 + x_3 \end{aligned}$$

The above example corresponds to the factorization $w = abb \cdot aca$. The "real" Parikh-matrices are

$$\begin{array}{ccc} abb \cdot aca & = & abbaca \\ \begin{pmatrix} 1 & 1 & 2 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} & = & \begin{pmatrix} 1 & 3 & 2 & 2 \\ 0 & 1 & 2 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Say that we also know the Parikh-vectors of another factorization $w = w_1'w_2'$ where $\Psi(w_1') = (1,1,0)$ and $\Psi(w_2') = (2,1,1)$ (in or example this corresponds to the factorization $w = ab \cdot baca$). We then obtain the following incomplete Parikh-matrices

$$\begin{pmatrix} 1 & 1 & x_1' & x_3' \\ 0 & 1 & 1 & x_2' \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 2 & y_1' & y_3' \\ 0 & 1 & 1 & y_2' \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 3 & z_1 & z_3 \\ 0 & 1 & 2 & z_2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where z_i obviously are the same as before. This gives us the following system of equations

$$\begin{aligned} z_1 &= y_1' + 1 + x_1' \\ z_2 &= y_2' + 1 + x_2' \\ z_3 &= y_3' + y_2' + x_1' + x_3' \end{aligned}$$

In our example the above incomplete Parikh-matrices really are the following

$$\begin{array}{ccc} ab \cdot baca & = & abbaca \\ \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} & = & \begin{pmatrix} 1 & 3 & 2 & 2 \\ 0 & 1 & 2 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Say that we also know the Parikh-vectors of a third factorization $w = \bar{w}_1\bar{w}_2\bar{w}_3$ with $\Psi(\bar{w}_1) = (1, 1, 0)$, $\Psi(\bar{w}_2) = (0, 1, 0)$ and $\Psi(\bar{w}_3) = (2, 0, 1)$. We then get the incomplete Parikh-matrices

$$\begin{pmatrix} 1 & 1 & u_1 & u_3 \\ 0 & 1 & 1 & u_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & v_1 & v_3 \\ 0 & 1 & 1 & v_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 2 & s_1 & s_3 \\ 0 & 1 & 0 & s_2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 3 & z_1 & z_3 \\ 0 & 1 & 2 & z_2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$
$$\begin{pmatrix} 1 & 1 & v_1 + 1 + u_1 & v_3 + v_2 + u_3 \\ 0 & 1 & 2 & v_2 + u_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 2 & s_1 & s_3 \\ 0 & 1 & 0 & s_2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 3 & z_1 & z_3 \\ 0 & 1 & 2 & z_2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

with the corresponding system of equations

$$\begin{aligned} z_1 &= s_1 + v_1 + 1 + u_1 \\ z_2 &= s_2 + 2 + v_2 + u_2 \\ z_3 &= s_3 + s_2 + v_1 + 1 + u_1 + v_3 + v_2 + u_3 \end{aligned}$$

and the corresponding "real" Parikh-matrices are

$$\begin{array}{ccc} ab \cdot b \cdot aca & = & abbaca \\ \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} & = & \begin{pmatrix} 1 & 3 & 2 & 2 \\ 0 & 1 & 2 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

The problem is to gather as much information as possible of w given the different factorizations of w together with the corresponding Parikh-vectors. In the above example the approach would be to determine z_1, z_2, z_3 as closely as possible. Probably by determine upper and lower bounds for these variables. One should take advantage of any special relations and bounds on x_i, x'_i, s_i, u_i and v_i given by that they are elements in Parikh-matrices.

One should note that even when the above Parikh-matrices are completed we are not able to determine w. The Parikh-matrices are, however, a convenient way of representing more of what is known about the order than what is possible by the Parikh mapping.

Then of course this is an example where $|\Sigma| = 3$. The problem is more complex for higher cardinalities.

References

[CK97] Christian Choffrut and Juhani Karhumäki. Combinatorics of words. In G. Rozenberg and A. Salomaa, editors, *Handbook on Formal Languages*, volume I. Springer, Berlin-Heidelberg-New York, 1997.

[Kau75] Arnold Kaufmann. *Theory of Fuzzy Subsets*, volume I. Academic Press, 1 edition, 1975.

[Lot97] M. Lothair, editor. *Combinatorics on Words*. Cambridge University Press, 2 edition, 1997.

[Lot02] M. Lothair, editor. *Algebraic Combinatorics on Words*. Cambridge University Press, 1 edition, 2002.

[MSSY00] Alexandru Mateescu, Arto Salomaa, Kai Salomaa, and Sheng Yu. On an extension of the parikh mapping. Technical Report TUCS-TR-364, Turku Centre for Computer Science, 6 2000.

[Zad65] Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.