Technische Universität Dresden
Fakultät Informatik

Diplomarbeit / Master Thesis

zum Thema / on the subject

Rational Forest Languages and Sequential Forest Transducers

zum Erreichen des Akademischen Grades / to receive the academic degree

Diplom-Informatiker / Master of Science

von / by

Lutz Straßburger
geboren am 21. April 1975 in Dresden

Eingereicht am 6. April 2000

Betreuer:	Prof. Dr.-Ing. habil. Heiko Vogler
Verantwortlicher Hochschullehrer:	Prof. Dr.-Ing. habil. Heiko Vogler
Zweiter Gutachter:	Prof. Dr. rer. nat. habil. Manfred Droste

Abstract

In the search for a "machine independent" characterization of the functions realized by deterministic tree transducers—similar to the characterization of generalized sequential machines (gsm)—it becomes necessary to consider tuples of trees instead of trees. As a consequence, some of the known results on recognizable tree languages can be extended. For that purpose, an algebraic structure, called *polyoid* will be introduced. The elements of the free polyoid are tuples of trees (which will be called *forests*) over a ranked alphabet Σ—in the same sense as the elements of the free monoid are words over an alphabet A.

As generalization of recognizable tree languages, the recognizable forest languages are introduced and studied. Furthermore, the rational forest languages are defined, and in the spirit of Kleene's theorem on the coincidence of the recognizable and rational languages in a free monoid, it will be shown that in the free polyoid, the recognizable and rational languages are the same. This result is a generalization of the result on regular tree languages by Thatcher and Wright.

The work will be concluded with a definition of *sequential forest transducers* and a "machine independent" characterization of the functions realized by them. The well-known characterization for gsm's will be a special case of that characterization.

Contents

1 Introduction

Kleene's theorem on the coincidence of the rational and the recognizable languages in a free monoid (cf. [Kle56]) is considered to be one of the cornerstones of theoretical computer science. Because of the great influence of that theorem to my work and because the main contribution of this thesis is a similar theorem for a more complex structure than the monoid, I will use a simple example to illustrate Kleene's theorem.

A language is called recognizable if it is recognized by some finite-state automaton (fsa). See [Eil76] or [Per90] for an introduction into the theory of finite automata. Consider for example the alphabet $A = \{a, b, c\}$ and an fsa with three states s, f and q, where s is an initial state and f is a final state. Let the state transitions be

$$\begin{aligned} s &\xrightarrow{a} f, \\ s &\xrightarrow{b} q, \\ q &\xrightarrow{b} q, \\ q &\xrightarrow{c} q, \text{ and} \\ q &\xrightarrow{c} f. \end{aligned}$$

They can be considered to be the edges of a graph, whose nodes are the states of the automaton (Figure 1). The language L recognized by this automaton is a subset of the set A^*, which is the

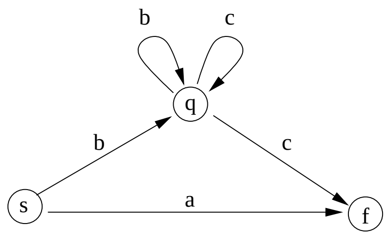

Figure 1: An fsa with three states.

set of all finite words over A. In our example, L contains the word a as well as all words that start with a b, end with a c and contain an arbitrary number of b's and c's in between.

A language is called rational if it can be represented by a rational expression, i.e. it can be built up from finite sets using only the operations $\cup$ (union), $\cdot$ (concatenation), and * (concatenation closure). The rational expression for the language in our example is

$$L \quad = \quad \{a\} \cup \{b\} \cdot \{b, c\}^* \cdot \{c\} \quad . \tag{1}$$

Note the obvious fact that there is just one concatenation operation (which is used twice in (1)) and that all subexpressions of (1), e.g. $\{b, c\}^*$, do represent subsets of A^*.

It is possible to define the recognizable as well as the rational sets in every monoid (e.g. [Eil76]), where Kleene's theorem does not hold in general. But there exist several generalizations of that theorem, for instance for trace monoids in [Och85], for formal power series in [Sch61], and for a combination of both in [DG99].

In [TW68] another important generalization is presented, namely the extention of the notion of finite-state automaton to that of finite-state tree automaton (fsta). This directly leads to the definition of recognizable tree languages. In opposition to an fsa, an fsta works on a ranked alphabet Σ, i.e. every symbol in Σ is equipped with a rank, which is a natural number. The language recognized by an fsta is a subset of the set T_Σ, which is the set of all finite trees over Σ.

Consider, for example, the ranked alphabet $\Sigma = \{\sigma, \gamma, \alpha\}$, where the symbol σ has rank 2, the symbol γ has rank 1, and α has rank 0. Now, define an fsta with two states p and f, where f is a final state. Observe that there are no initial states. Further, the state transitions cannot be drawn as a graph because they represent the edges of a hyper-graph. In the literature (e.g. [GS97]), they are represented as rules of a term rewriting system. For example

$$\begin{array}{rcl} \alpha & \to & p, \\ \alpha & \to & f, \\ \gamma(p) & \to & p, \text{ and} \\ \sigma(p, f) & \to & f. \end{array}$$

The tree language L accepted by this automaton contains the tree α and all trees that have the form $\sigma(\gamma \ldots \gamma\alpha, \sigma(\gamma \ldots \gamma\alpha, \sigma(\ldots, \sigma(\gamma \ldots \gamma\alpha, \alpha) \ldots)))$, which is more vivid if it is drawn as a tree (see Figure 2), where the number of γ's in each chain and the number of σ's can be any natural number (including zero).

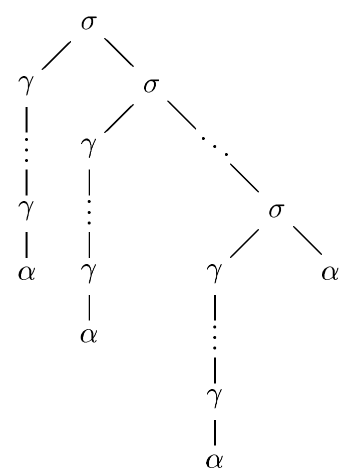

Figure 2: The form of the trees in language L.

In order to find for this language a similar rational expression as in (1), we encounter the problem that there is no straightforward concatenation operation for trees. The concatenation of two words is simply their juxtaposition. But what is the concatenation of two trees? The only possibility is to paste one tree into some leaves of the other tree. In order to decide into which leaves of the first tree the second tree has to be plugged in, we have to name them. In a tree over a ranked alphabet Σ, the leaves are labeled by the 0-ary symbols of Σ. Therefore, for every 0-ary symbol of Σ, another concatenation operation has to be defined. Furthermore, in order to obtain a rational expression for every recognizable tree language, the ranked alphabet Σ has to be extended. In [TW68], this has been worked out in detail. The rational expression for the language in our example is

$$L \quad = \quad \{\alpha\} \cup (\{\sigma(p,f)\} \cdot_p \{\gamma(p)\}^{*p} \cdot_p \{\alpha\})^{*f} \cdot_f \{\alpha\} \quad . \tag{2}$$

Although the language L is a subset of T_Σ, the sets $\{\sigma(p,f)\}$ and $\{\sigma(p,f)\} \cdot_p \{\gamma(p)\}^{*p}$, that occur in in (2), are not subsets of T_Σ but subsets of $T_{\Sigma\cup\{p,f\}}$, where the rank of p and f is zero. There occur two different concatenation operations in (2) because there are two states in the automaton.

It has been shown in [TW68], that for every recognizable tree language there exists such a rational expression, and that every tree language that can be represented by a rational expression is indeed recognizable. This result has been an important achievement because it shows that the notion of finite-state automaton as well as the notion of rational expression can be extended from words to trees, and that the recognizable tree languages have a similar characterization as the recognizable languages in the free monoid. For this reason, the work by Thatcher an Wright is of great significance for the development of the theory of recognizable tree languages as we know it now. However, there are two drawbacks:

(i) The language in which the rational expressions are formulated is not predefined. For a given recognizable tree language $L \subseteq T_\Sigma$, the number of different concatenation operations that occur in the rational expression for L is determined by the automaton that recognizes the language L. But in a rational expression for word languages only one concatenation operations may occur.

(ii) The tree languages that are obtained by evaluating a subexpression of a rational expression for a tree language $L \subseteq T_\Sigma$ are usually not subsets of T_Σ. However, in the case of word languages, a subexpression of a rational expression (for a language $L \subseteq A^*$) always represents a subset of A^*.

In the literature (e.g. [GS97]) we can also find a slightly different approach. Instead of considering tree languages $L \subseteq T_\Sigma$ and defining a concatenation operation for every 0-ary symbol in Σ, we can consider tree languages $L \subseteq T_\Sigma(X)$, where $T_\Sigma(X)$ is the set of all finite trees over the ranked alphabet Σ and a finite set X of variables. Then, for every variable in X a concatenation operation is defined. Observer that this does solve neither of the two problems mentioned above, because in order to find a rational expression for a given recognizable tree language $L \subseteq T_\Sigma(X)$, we have to introduce another finite set Z of new variable symbols and define a concatenation operation for each of them. The tree languages that occur as subexpressions of the rational expression for L are no longer subsets of $T_\Sigma(X)$ but of $T_\Sigma(X \cup Z)$.

The reason of this behavior of trees lies in their ability of branching. It has already been observed in [TW68] that the extension of the ranked alphabet Σ cannot be avoided.

However, let me ask the question, whether the extension of the alphabet can be avoided by choosing an alternative approach to the notion of rationality. For instance, from a pragmatic point of view, it might be reasonable to insist on having just one concatenation operation instead of many. In that case, we have to change the nature of the concatenation in such a way that many tree concatenation operations are performed in parallel. But then we cannot concatenate just two trees, but we have to concatenate one tree with a tuple of trees. This means that the operation becomes asymmetric (in the sense that the first operand is a tree, the second operand is a tuple of trees and the result is again a tree) and therefore it does not make sense to ask for properties like associativity. The solution to this problem is to consider only tuples of trees. I will call those tuples *forests*. The set of all forests over a ranked alphabet Σ will be denoted by $\Sigma^{\circledast}$. Let me informally sketch out the idea of forest. Seen from the inside, a forest is a tuple of trees. Seen from the outside, it can be described as a Lego brick. Consider again, the ranked alphabet $\Sigma = \{\sigma, \gamma, \alpha\}$ as in the example above. Then Figure 3 shows a forest that contains two trees. Let me call it s.

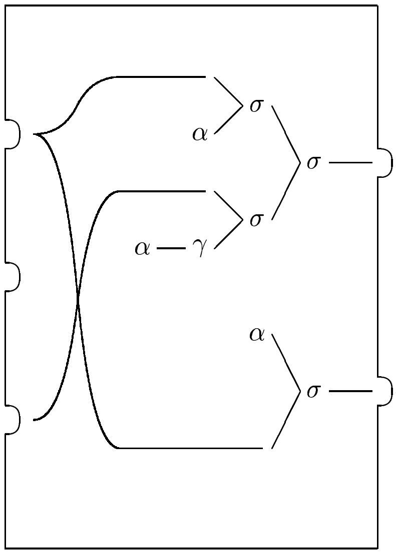

Figure 3: The forest $s = \langle\langle\langle\pi^3_1, \alpha\rangle\sigma, \langle\pi^3_3, \alpha\gamma\rangle\sigma\rangle\sigma, \langle\alpha, \pi^3_1\rangle\sigma\rangle \in \Sigma^{\circledast}$

The forest s will be denoted by $\langle\langle\langle\pi^3_1, \alpha\rangle\sigma, \langle\pi^3_3, \alpha\gamma\rangle\sigma\rangle\sigma, \langle\alpha, \pi^3_1\rangle\sigma\rangle$. The symbols π^3_1 and π^3_3 do not occur in the figure. They are represented by the connections between the holes on the lefthand-side of the box and the trees in the box. The number 3 in the superscript represents the total number of holes in the box. The subscript stands for the number of the hole to which the open position in the tree is connected. Because the symbols π^3_1 and π^3_3 represent trees without any symbols from the alphabet Σ, they will be called *empty trees*, but for the moment they can be seen as variables. In other words, all variables that occur in a forest have to have the same n in the superscript and an $i = 1, \ldots, n$ in the subscript. The forest s in our example has potentially three different variables, which is expressed by the three holes on the lefthand-side of the box. The first variable occurs twice, the third once, and the second does not occur at all. Since s contains two trees, there are two plugs on the righthand-side of the box. For that reason, it is called a $(3, 2)$-forest. Another example is the forest $t = \langle\langle\pi^2_2, \pi^2_2\rangle\sigma, \langle\alpha, \pi^2_1\rangle\sigma\rangle\sigma$, which is shown in Figure 4.

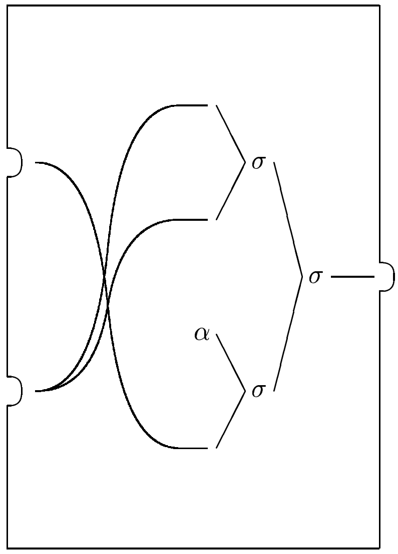

Figure 4: The forest $t = \langle\langle\pi_2^2, \pi_2^2\rangle\sigma, \langle\alpha, \pi_1^2\rangle\sigma\rangle\sigma \in \Sigma^{\circledast}$

The forest t is a $(2,1)$-forest. Informally speaking, the forests s and t can be plugged together because s has two plugs and t has two holes. Formally, this operation is called concatenation. The result of the concatenation of s and t is denoted by $s;t$. In our example, $s;t = \langle\langle\langle\alpha, \pi_1^3\rangle\sigma, \langle\alpha, \pi_1^3\rangle\sigma\rangle\sigma, \langle\alpha, \langle\langle\pi_1^3, \alpha\rangle\sigma, \langle\pi_3^3, \alpha\gamma\rangle\sigma\rangle\sigma\rangle\sigma\rangle\sigma$, which is shown in Figure 5.

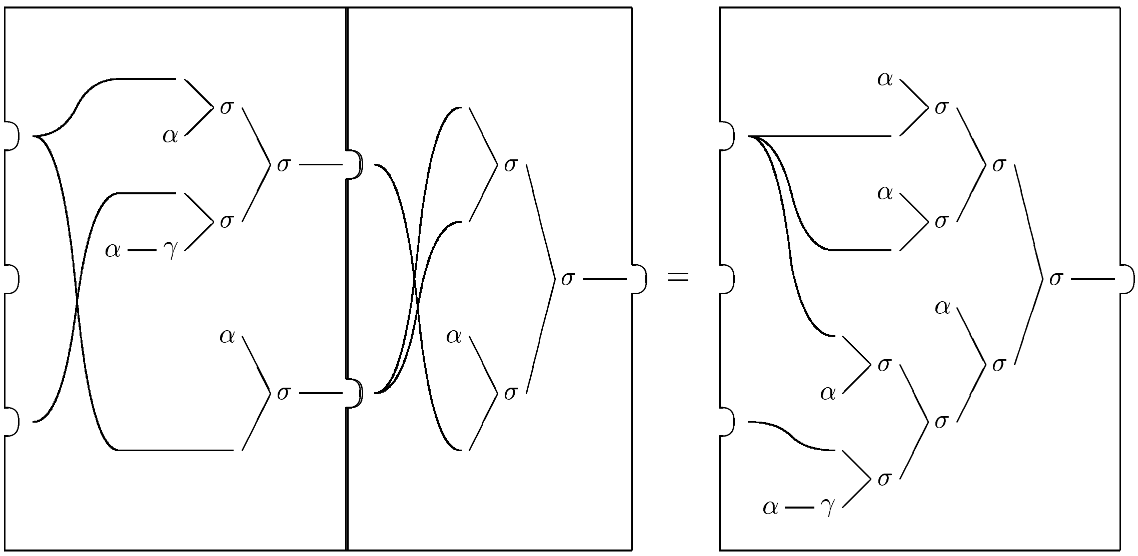

Figure 5: The concatenation of s and t.

The forest $s;t$ is a $(3,1)$-forest. This example illustrates the main idea behind the concatenation operation. Two forests s and t can be concatenated if there are natural numbers n, m, k such that s is an (n,m)-forest and t is an (m,k)-forest. Then, the results $s;t$ is an (n,k)-forest.

With this approach, the many tree concatenation operations have been reduced to a single forest concatenation operation. Note, that the new concatenation operation is a partial operation on $\Sigma^{\circledast}$, whereas the concatenation operation for words is total. But I do not consider this to be a problem because any partial operation can be extended to a total operation.

It is possible to define the recognizable forest languages as generalizations of the recognizable tree languages. The basis is the notion of finite-state forest automaton (fsfa), which is a generalization of the notion of fsta. The concept of rational forest languages can be defined by means of rational expression. The only operations that can occur in a rational expression are the union $\cup$, the cartesian product $\times$, the forest language concatenation $\cdot$, and the forest language concatenation closure *. The forest language concatenation $\cdot$ is an extension of the forest concatenation operation ; to sets of forests. Later, it will become clear, why I use different symbols for both operations. The forest language concatenation closure * is, as the name suggests, the closure operation for the forest language concatenation—in the same sense as the Kleene star * is the closure operation for the word concatenation.

Let me stress the fact that there is only one concatenation operation and only one star operation, but, in opposition to word languages and tree languages, there is an additional operation $\times$ to obtain the cartesian product of two forest languages. Informally speaking, the reason for the necessity of the product is that the set $\Sigma^{\circledast}$ forms a cartesian category and the product is an essential part of cartesian categories.

It should be clear now why I believe that considering forests instead of trees is a solution to the first problem mentioned above because there is no need of extending the alphabet in the rational expression for a recognizable forest language. But forests do also quite naturally help to work around the second problem. Whenever we form a rational expression for a forest language, we do not leave the set $\Sigma^{\circledast}$.

As an example let me use the same language L as above. This is possible because the set T_Σ is isomorphic to the set of all $(0,1)$-forests in $\Sigma^{\circledast}$, i.e. all forests that consist of one tree that contains no variables. Figure 6 shows, how the language L can be depicted as $(0,1)$-forest language.

The new rational expression is

$$L \;=\; \{\alpha\} \cup \big(\{\alpha\} \cdot \{\pi_1^1\gamma\}^\star \times \{\alpha\}\big) \cdot \{\langle \pi_1^2, \langle \pi_1^2, \pi_2^2\rangle\sigma\rangle\}^\star \cdot \{\langle \pi_1^2, \pi_2^2\rangle\sigma\}, \tag{3}$$

which is depicted in Figure 7. Observe that there is only one concatenation operation and that every subexpression of (3) represents a subset of $\Sigma^{\circledast}$.

At this point, the attentive reader might be concerned whether the use of forests instead of trees is indeed an improvement because

(i) the problem of having many tree concatenation operations is solved by defining one partial forest concatenation operation which essentially performs many tree concatenation operations at the same time, and

(ii) the problem of extending the alphabet by an arbitrary large number of constant symbols is solved by introducing an infinite number of variable symbols as part of the forest structure.

However, there are two main reasons why I propose to use forests instead of trees. The first is symmetry and the second is utility.

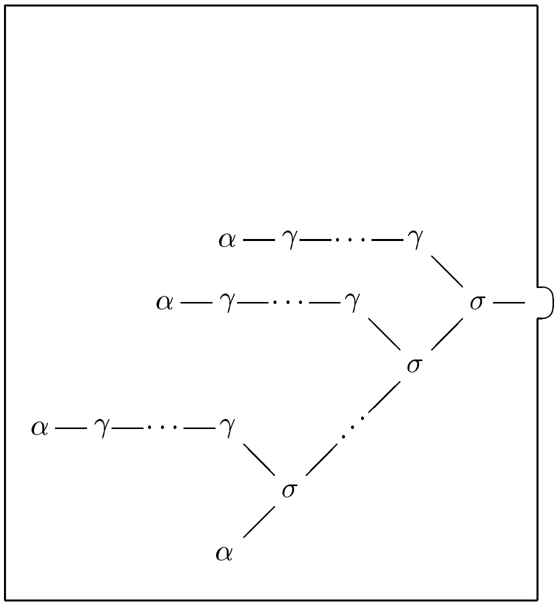

Figure 6: The form of the forests in language L.

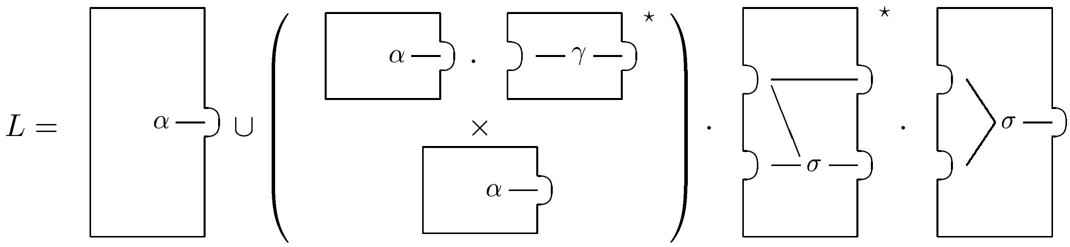

Figure 7: The rational expression in (3). The braces are omitted for better readability.

Let me first explain what I mean by symmetry. Consider again Kleene's original theorem, which acts on the set A^* of all words over an alphabet A. The set A^* forms a monoid, namely the free monoid generated by A. This will be made precise in Section 2. The concatenation operation for words (that is used in the definition of rational languages) is exactly the binary operation of the monoid. As a consequence, it is straightforward to extend the definitions for recognizable and rational languages to all monoids (cf. [Eil76]).

Let us now look at the algebraic structure that is formed by the set T_Σ of all trees over a ranked alphabet Σ. Obviously, it is a Σ-algebra, namely, the free Σ-algebra generated by the empty set—For a given set X of variable symbols, the set $T_\Sigma(X)$ forms the free Σ-algebra generated by X. I will make this precise in Section 3. The tree concatenation operation, as defined above, has nothing to do with the Σ-algebra. For that reason, it does not make sense to define rational languages in any Σ-algebra except for the free Σ-algebra.

Furthermore, the definition of a monoid is totally independent of any alphabet A, whereas the definition of a Σ-algebra depends on the ranked alphabet Σ.

This is an asymmetry between words and trees. In order regain a symmetry, we have to find an algebraic structure—let me call it *polyoid*—with a binary operation, such that the set $\Sigma^{\circledast}$ of all forests over a ranked alphabet Σ forms the free polyoid generated by Σ, and such that the forest concatenation operation is the binary operation of that polyoid—in the same sense as the word concatenation operation is the binary operation of the free monoid.

A polyoid is a special case of a cartesian category—in the same sense as a monoid can be seen as a category with a single object. More precisely, if we consider the set $\Sigma^{\circledast}$ of all forests over Σ as a category with the natural numbers as objects and for every $n, m \in \mathbb{N}$, the (n, m)-forests as morphisms between n and m, then this category is isomorphic to the algebraic theory $\mathbf{Th}(\Sigma)$ over Σ (cf. [Law63]). However, in this thesis, I prefer not to introduce category theory nor to demand category theory as a prerequisite for the reading. In Section 4, I will introduce polyoids as an algebraic structure and show that the set $\Sigma^{\circledast}$ of all forests over a ranked alphabet Σ is indeed the free polyoid generated by Σ.

In order to summarize the previous paragraphs, Table 1 shows an overview over the contents of Sections 2, 3, and 4. Observe the asymmetry between the first and the second column, as well as the symmetry between the first and the third column.

Section 2	Section 3	Section 4
alphabet A	ranked alphabet Σ set of variables X	ranked alphabet Σ
$\Downarrow$	$\Downarrow$	$\Downarrow$
set A^* of all words over A	set $T_\Sigma(X)$ of all Σ-trees over X	set $\Sigma^{\circledast}$ of all forests over Σ
$\Uparrow$	$\Uparrow$	$\Uparrow$
free monoid generated by A	free Σ-algebra generated by X	free polyoid generated by Σ

Table 1: The contents of Sections 2, 3, and 4.

In Sections 5, 6, and 7, the recognizable as well as the rational forest languages are introduced and studied. The main result of those sections will be that the set $Rec(\Sigma^{\circledast})$ of all recognizable forest languages in $\Sigma^{\circledast}$ is equal to the set $Rat(\Sigma^{\circledast})$ of all rational forest languages in $\Sigma^{\circledast}$.

Let me now explain what I mean by utility. Kleene's theorem has many applications in all fields of theoretical computer science, for example in finding "machine independent" characterizations for transductions realized by transducers, which are basically fsa's equipped with an output (see [Ber79, ch.III] for an overview). One class of transducers are the so called sequential transducers, also known as generalized sequential machines (gsm). A gsm consists of a set of states Q, an input alphabet A, an output alphabet B, an initial state $q_0 \in Q$, a state transition function $\delta : Q \times A \to Q$ and an output function $\lambda : Q \times A \to B^*$. Without going into further details, which can be found in [Ber79, ch.IV], let me say that a gsm realizes a function $f : A^* \to B^*$. In [GR66], a "machine independent" characterization of the functions realized by gsm's is presented. Namely, a function $f : A^* \to B^*$ is realized by a gsm if and only if

(i) $f(\varepsilon) = \varepsilon$, where ε denotes the empty word in A^* and B^*,

(ii) for every $w \in A^*$ and $a \in A$, there is a $v \in B^*$ such that

$$f(wa) = f(w) \cdot v,$$

(iii) there is a natural number z such that for every $w \in A^*$ and $a \in A$,

$$\mathit{length}(f(wa)) < \mathit{length}(f(w)) + z,$$

(iv) for every rational language $L \in B^*$, the languages $f^{-1}(L)$ is also rational.

In order to obtain this characterization, we have to use Kleene's theorem, as well as some properties of the concatenation operation, which follow from the fact that A^* is a free monoid.

For tree languages, a similar notion of transducer exists. In the literature (e.g. [Eng75], [Eng77], [FV98]) we can find a large number of different classes of tree transducers. Roughly speaking, they can be divided into two main classes: top-down tree transducers (tdtt) and bottom-up tree transducers (butt). A tree transducer consists of a set of states Q, an input alphabet Σ, an output alphabet Ω, a distinct set of states $Q_d \subseteq Q$, and a set R of rules. In the case of a tdtt, the set Q_d contains the initial states and in the case of a butt, it contains the final states. The rules in R are rules of a term rewriting system and contain information about the state transition and the output. I will not got into further details, which can be found in [Eng75]. Important is the fact that if such a tree transducer is deterministic, then it realizes a (partial) function $f : T_\Sigma \to T_\Omega$.

I am not aware of a "machine independent" characterization of the functions realized by any class of tree transducers. In the attempt to obtain a naive extension of the characterization of gsm's for certain classes of tree transducers, I encountered two problems. First, the result of [TW68] is not as easy to handle as Kleene's theorem, and second, the tree concatenation operation has not the same nice properties as the word concatenation operation. This makes it impossible to mimic the construction in [GR66].

However, it is straightforward to extend any tree transducer to a forest transducer. In Section 9, I will define the notion of *deterministic bottom-up forest transducer* (*dbuft*), which will be a generalization of the notion of deterministic butt, with the restriction that there are no final states. Furthermore, I will show a "machine independent" characterization for the functions realized by a subclass of dbuft's. Because of the similarity to the result in [GR66] and because that result is a special case of mine, I will call this class *sequential forest transducers* (*seqft*). In terms of [Eng75] and [GS97], a seqft is a deterministic, totally defined, linear, and non-deleting butt that has no final states and that has the additional property that the variables on the right-hand side of the rules must not permute.

In order to obtain a "machine independent" characterization of those transducers in Section 9, we have to make use of the notion of forest. Furthermore, the theorem on the coincidence of the recognizable and the rational forest languages plays an essential role. One reason why forests are unavoidable in this construction is the necessity of another class of forest languages, namely the semi-recognizable forest languages. I will introduce them in Section 8. It does not make sense to define this class for trees languages because in the case of trees the semi-recognizable languages and the recognizable languages coincide by definition.

This shows that the use of forests instead of trees not only restores a broken symmetry, but also helps to obtain a butter understanding of well-known constructions on trees.

2 Preliminaries

This section is concerned with some preliminary mathematical concepts needed for the introduction of my notion of forests in Section 4. The main topics are the clarification of my notation and terminology, and the introduction of monoids and free monoids. I will also discuss some fine points, that might cause trouble if not addressed early.

2.1 Sets I assume the reader to be acquainted with the mathematical concepts of *sets*, *relations*, and *functions*. A very good introduction to the foundations of axiomatic set theory is the first chapter of [Kun80]. Also, the first chapter of [BW99] can be used as a very good and easy readable introduction to the basic notions of sets and functions.
If A and B are sets, then $A \subseteq B$ means "A is a subset of B", and $A \subset B$ means "A is a proper subset of B". Further, $A \cup B$, $A \cap B$, and $A \setminus B$ denote their *union*, *intersection* and *set difference*, respectively.
If I is a set and $\mathcal{F} = \{A_i \mid i \in I\}$ is a family of sets indexed by I, then

$$\bigcup_{i \in I} A_i = \{x \mid \exists i \in I \;.\; x \in A_i\} \quad \text{and} \quad \bigcap_{i \in I} A_i = \{x \mid \forall i \in I \;.\; x \in A_i\}$$

denote the *union* and *intersection* (respectively) of the sets in $\mathcal{F}$.
The set $\mathbb{N}$ denotes the set of natural numbers including the zero, i.e. $\mathbb{N} = \{0, 1, 2, 3, \dots\}$.
For a given set A, the set $\mathfrak{P}(A) = \{C \mid C \subseteq A\}$ denotes the *power set* (the set of all subsets) of A.
For two sets A and B, the set $A \times B = \{\langle a, b\rangle \mid a \in A \wedge b \in B\}$ denotes the *cartesian product* of A and B. One should be aware of the fact that for three given sets A, B and C, the two sets $(A \times B) \times C$ and $A \times (B \times C)$ are not equal. But because there is an obvious bijection, both sets can be identified. More general, if $n \in \mathbb{N}$ and $A_1, \dots, A_n$ are sets, then

$$A_1 \times \dots \times A_n = \{\langle a_1, \dots, a_n\rangle \mid a_1 \in A_1 \wedge \ldots \wedge a_n \in A_n\}$$

is the cartesian product of the sets $A_1, \dots, A_n$. Further, for every set A and natural number n,

$$A^n = \underbrace{A \times \dots \times A}_{n \text{ times}} = \{\langle a_1, \dots, a_n\rangle \mid a_1 \in A \wedge \ldots \wedge a_n \in A\}$$

is the set of all n-tuples of elements of A. By definition, for every set A, the set $A^0 = \{\langle\rangle\}$ is a singleton, where $\langle\rangle$ denotes the empty tuple.

2.2 Functions If A and B are sets, then $f : A \to B$ denotes a function f with *domain* A and *codomain* B. I will write $f : a \mapsto b$ or $f(a) = b$ to say that b is the value of f at a. For example,

$$succ : \mathbb{N} \to \mathbb{N} : n \mapsto n + 1$$

denotes the successor function for the natural numbers. I could also write $succ(n) = n + 1$.
If $f : A \to B$ is a function from A to B, then the set $im(f) = \{b \in B \mid \exists a \in A . f(a) = b\}$ is called the *image* of f. In the successor example, $im(succ) = \mathbb{N} \setminus \{0\}$.
For an element $b \in B$, the set $f^{-1}(b) = \{a \in A \mid f(a) = b\}$ is called the *inverse image* of b. Similarly, for a subset $B' \subseteq B$, the set $f^{-1}(B') = \{a \in A \mid \exists b \in B' \;.\; f(a) = b\} = \bigcup_{b \in B'} f^{-1}(b)$ is called the *inverse image* of B'.

If A is a set and $k \in \mathbb{N}$, then a function $f : A^k \to A$ will be called a *k-ary operation on A*. A 0-ary operation $f : A^0 \to A$ on A is a constant, namely the value of f at $\langle\rangle$.
A binary operation f on a set A is called *partial* if is defined only on a subset of $A \times A$. It will be denoted by $f : A \times A \dashrightarrow A$. For example, the subtraction on the natural numbers is a partial operation because it is not defined in the natural numbers if the second argument is bigger than the first one.

2.3 Definition Let I be a set. An *I-indexed set* is a set A together with a function $f : A \to I$. If $a \in A$ and $f(a) = i$, then *a has index i*.

2.4 If A is an I-indexed set, then for every $i \in I$, the set $A_i = f^{-1}(i)$ contains all elements of A, that have index i. It is obvious, that

$$A = \bigcup_{i \in I} A_i. \tag{4}$$

For every I-indexed set A, it is always the case that

$$\forall i, j \in I \ . \ i \neq j \implies A_i \cap A_j = \emptyset. \tag{5}$$

Therefore, in order to define an I-indexed set A, it is sufficient to define for every $i \in I$ the set A_i without naming the index function $f : A \to I$ explicitely.
Sometimes the discussion is casted in terms of a family $\mathcal{F} = \{A_i \mid i \in I\}$ indexed by the set I (as in 2.1). However, when I refer to an I-indexed set A, then (5) does hold, but for a family $\mathcal{F} = \{A_i \mid i \in I\}$ of sets indexed by I, this is not necessarily true.

2.5 Remark In the remainder of this section, I will define the notions of monoids, monoid homomorphisms, and free monoids. I will do this very thoroughly and I hope the reader will not be deterred. The reason for the meticulous elaboration is the fact that I will do a similar elaboration for a more complex algebraic structure in Section 4, which will be easier to understand with the help of this section.

2.6 Definition A *monoid* $\mathcal{M} = \langle M, \circ_{\mathcal{M}}, \varepsilon_{\mathcal{M}} \rangle$ is a triple consisting of

- a set M (called the *underlying set of $\mathcal{M}$*),
- a binary operation $\circ_{\mathcal{M}} : M \times M \to M$, and
- a distinguished element $\varepsilon_{\mathcal{M}} \in M$ (called *identity* or *unit element*),

such that

(i) for every $a \in M$, $a \circ_{\mathcal{M}} \varepsilon_{\mathcal{M}} = a = \varepsilon_{\mathcal{M}} \circ_{\mathcal{M}} a$, and

(ii) for every $a, b, c \in M$, $(a \circ_{\mathcal{M}} b) \circ_{\mathcal{M}} c = a \circ_{\mathcal{M}} (b \circ_{\mathcal{M}} c)$.

The first equation justifies the name "unit element" for $\varepsilon_{\mathcal{M}}$, and the second one says that the operation $\circ_{\mathcal{M}}$ is associative.

2.7 Remark For my work, I consider it to be important to distinguish between the monoid $\mathcal{M}$ and its underlying set M. As a notational convention, a calligraphic letter $\mathcal{M}$ or $\mathcal{N}$ will denote the monoid, whereas the according Roman letter M or N denotes its underlying set. The same convention will be used in later sections for other algebraic structures.

2.8 Example The natural numbers form together with the addition operation and the number $0 \in \mathbb{N}$ as unit element a monoid $\mathcal{M}_1 = \langle \mathbb{N}, +, 0 \rangle$, and together with the multiplication operation and the number $1 \in \mathbb{N}$ as unit element a monoid $\mathcal{M}_2 = \langle \mathbb{N}, \cdot, 1 \rangle$. Although both monoids have the same underlying set $M_1 = M_2 = \mathbb{N}$, they are certainly not equal (nor isomorphic).

2.9 Definition Let $\mathcal{M} = \langle M, \circ_{\mathcal{M}}, \varepsilon_{\mathcal{M}} \rangle$ and $\mathcal{N} = \langle N, \circ_{\mathcal{N}}, \varepsilon_{\mathcal{N}} \rangle$ be monoids. Further, let $h : M \to N$ be a function that preserves the monoid structure, i.e.

$$\begin{aligned} h(\varepsilon_{\mathcal{M}}) &= \varepsilon_{\mathcal{N}}, \quad \text{and} \\ h(a \circ_{\mathcal{M}} b) &= h(a) \circ_{\mathcal{N}} h(b), \quad \text{for every } a, b \in M. \end{aligned}$$

Then $h : M \to N$ is called a *monoid homomorphism* and will be written as $h : \mathcal{M} \to \mathcal{N}$.

2.10 Definition Let $\mathcal{M} = \langle M, \circ_{\mathcal{M}}, \varepsilon_{\mathcal{M}} \rangle$ and $\mathcal{N} = \langle N, \circ_{\mathcal{N}}, \varepsilon_{\mathcal{N}} \rangle$ be monoids, and $h : \mathcal{M} \to \mathcal{N}$ be a monoid homomorphism. Then, h is called a *monoid isomorphism* if there is a monoid homomorphism $g : \mathcal{N} \to \mathcal{M}$ such that for every $m \in M$ and every $n \in N$,

$$g(h(m)) = m \qquad \text{and} \qquad h(g(n)) = n.$$

Two monoids $\mathcal{M}$ and $\mathcal{N}$ are called *isomorphic* iff there is a monoid isomorphism $h : \mathcal{M} \to \mathcal{N}$.

2.11 Remark Every monoid homomorphism which is a bijection is a monoid isomorphism.

2.12 Definition Let A be a finite set. The set A^* of all (finite) *words* over A is defined to be the smallest set such that

(i) there is an element $\varepsilon \in A^*$ (called the *empty word*), and

(ii) for every $a \in A$ and every word $w \in A^*$, there is a word $wa \in A^*$.

2.13 Example Let $A = \{a, b, c, d\}$. Then, following the definition literally, the word $\varepsilon accccb$ is an element of A^*. However, the ε in the beginning is usually omitted because it is a simply a sign to denote the empty word. This means that $accccb \in A^*$.

2.14 For a given alphabet A, the *word concatenation operation* $\cdot : A^* \times A^* \to A^*$ is the juxtaposition of two words, for example (where A is as in Example 2.13)

$$abcc \cdot cdca = abcccdca.$$

Formally, this binary operation can be defined inductively as follows.

(i) For every $w \in A^*$,

$$w \cdot \varepsilon = w,$$

(ii) for every $v, w \in A^*$ and $a \in A$,

$$v \cdot (wa) = (v \cdot w)a.$$

2.15 Definition Let A be a set. Then, the monoid $F_{\text{Mon}}A = \langle A^*, \cdot, \varepsilon \rangle$, where the underlying set A^* is the set of all words over A, the operation $\cdot$ is the word concatenation operation

$$\cdot : A^* \times A^* \to A^* : \langle s, t \rangle \mapsto st,$$

and the unit element $\varepsilon \in A^*$ is the empty word, is called the *free monoid* generated by A.

2.16 Remark The name "free monoid" will be justified by Proposition 2.17, which says that for every set A and every monoid $\mathcal{M} = \langle M, \circ_{\mathcal{M}}, \varepsilon_{\mathcal{M}} \rangle$, any function $g : A \to M$ can uniquely be extended to a monoid homomorphism $\hat{g} : F_{\text{Mon}}A \to \mathcal{M}$. For many mathematical structures, this property is the characterization of the corresponding "free structure", which is not unique. But it is unique up to isomorphism. In the case of the monoid for instance, we could define the monoid $F'A = \langle \mathit{list}A, \cdot, [] \rangle$, where the underlying set $\mathit{list}A$ is the set of all finite lists of elements of A (e.g. $[a, c, c, b] \in \mathit{list}\{a, b, c, d\}$), the binary operation $\cdot : \mathit{list}A \times \mathit{list}A \to \mathit{list}A$ is the list concatenation, and the unit element $[]$ is the the empty list. Then $F'A$ and $F_{\text{Mon}}A$ are isomorphic, but not equal.

2.17 Proposition *Let A be a set and $\mathcal{M} = \langle M, \circ_{\mathcal{M}}, \varepsilon_{\mathcal{M}} \rangle$ be a monoid. Then, for every function $g : A \to M$, there is a unique monoid homomorphism $\hat{g} : F_{\text{Mon}}A \to \mathcal{M}$, such that for every $a \in A$,*

$$\hat{g}(a) = g(a).$$

Proof: The proof is well-known. Define $\hat{g} : F_{\text{Mon}}A \to \mathcal{M}$ as follows. Let $w \in A^*$. If $w = \varepsilon$, then let $\hat{g}(w) = \varepsilon_{\mathcal{M}}$. Otherwise, $w = a_1a_2 \ldots a_n$, for some $n \geq 1$ and $a_1, a_2, \ldots, a_n \in A$. Then let

$$\hat{g}(w) = g(a_1) \circ_{\mathcal{M}} g(a_1) \circ_{\mathcal{M}} \ldots \circ_{\mathcal{M}} g(a_n).$$

It immediately follows that $\hat{g}$ is a homomorphism and that for every $a \in A$, $\hat{g}(a) = g(a)$. This shows that $\hat{g}$ exists. To show that it is unique, suppose there is a homomorphism $\hat{g}' : F_{\text{Mon}}A \to \mathcal{M}$ with $\hat{g}'(a) = g(a)$ for every $a \in A$. Then $\hat{g}'(\varepsilon) = \varepsilon_{\mathcal{M}} = \hat{g}(\varepsilon)$ because $\hat{g}'$ is an homomorphism. Furthermore, for every $n \geq 1$ and $a_1, a_2, \ldots, a_n \in A$,

$$\begin{aligned} \hat{g}'(a_1a_2 \ldots a_n) &= g(a_1) \circ_{\mathcal{M}} g(a_2) \circ_{\mathcal{M}} \ldots \circ_{\mathcal{M}} g(a_n) \\ &= \hat{g}(a_1a_2 \ldots a_n). \end{aligned}$$

Hence, $\hat{g}' = \hat{g}$. □

3 Trees and Σ-Algebras

3.1 In this section, I will introduce the notion of trees over ranked alphabets as it can be found in most of the literature (e.g. [FV98] or [Wec92]).

3.2 Definition A *ranked alphabet* Σ is a finite $\mathbb{N}$-indexed set of symbols.

3.3 In other words, if Σ is a ranked alphabet, then there is a function $\mathit{rank} : \Sigma \to \mathbb{N}$, which assigns to every symbol $\sigma \in \Sigma$ its rank. For a given natural number $k \in \mathbb{N}$, the set

$$\Sigma_k = \{\sigma \in \Sigma \mid \mathit{rank}(\sigma) = k\} = \mathit{rank}^{-1}(k)$$

contains all symbols in Σ that have rank k. Furthermore,

$$\Sigma = \bigcup_{k \geq 0} \Sigma_k.$$

3.4 Remark Since a ranked alphabet Σ is finite, there will always be a number $maxrank(\Sigma) \in \mathbb{N}$ such that for every $k > maxrank(\Sigma)$, we have $\Sigma_k = \emptyset$. Throughout this thesis, statements like "for every $k \geq 0 \ldots$ " will occur, which will become vacuous for those k with $\Sigma_k = \emptyset$. For this reason, I will not mention $maxrank(\Sigma)$ anymore. The reader should just keep in mind that ranked alphabets are always finite, and therefore, for almost all $k \in \mathbb{N}$, $\Sigma_k = \emptyset$.

3.5 Definition Let Σ be a ranked alphabet and X be a set of variable symbols (such that $\Sigma \cap X = \emptyset$). Then, the set $T_\Sigma(X) \subseteq (\Sigma \cup X \cup \{(\} \cup \{)\} \cup \{,\})^*$ of *trees over* Σ *and* X (or Σ*-trees over* X) is the smallest set T, such that

- $X \subseteq T$ and
- if $\sigma \in \Sigma_k$ with $k \geq 0$, and $t_1, \ldots, t_k \in T$, then $\sigma(t_1, \ldots, t_k) \in T$.

3.6 Whenever I use a set X of variables together with a ranked alphabet Σ, I assume Σ and X to be disjoint. This should not cause problems, because variables can always be renamed.

3.7 Usually, the set of variables is finite. If $n \geq 0$ is a natural number, then the set $X_n = \{x_1, x_2, \ldots, x_n\}$ denotes a finite set of n variable symbols. By definition, put $X_0 = \emptyset$.

3.8 Example Consider the ranked alphabet $\Sigma = \Sigma_0 \cup \Sigma_2$ with $\Sigma_0 = \{\alpha\}$ and $\Sigma_2 = \{\sigma\}$. Then, some elements of the set $T_\Sigma(X_3)$ are, for example, the trees x_1, x_2, x_3, $\alpha()$, $\sigma(x_1, \alpha())$, and $\sigma(\sigma(x_2, x_3), \sigma(\sigma(x_1, x_2)))$.

3.9 If Σ is a ranked alphabet and $n, m \in \mathbb{N}$, then the set

$$T_\Sigma(X_n)^m = \underbrace{T_\Sigma(X_n) \times \ldots \times T_\Sigma(X_n)}_{m \text{ times}}$$

denotes the set of all m-tuples of Σ-trees over n variables. If $m = 0$, then $T_\Sigma(X_n)^m$ is a singleton and contains only the empty tuple $\langle\rangle$ (see 2.1).

3.10 Definition Let A be a set and Σ be a ranked alphabet. Furthermore, let $k \in \mathbb{N}$ and $\sigma \in \Sigma_k$ be a symbol of rank k. A *realization* of σ in A is a function $\sigma^A : A^k \to A$.

3.11 Definition Let Σ be a ranked alphabet. A Σ*-algebra* $\mathcal{A} = \langle A, \Sigma^A \rangle$ is a pair consisting of a set A, called the *underlying set* of $\mathcal{A}$, and a set $\Sigma^A = \{\sigma^A \mid \sigma \in \Sigma\}$ of realizations σ^A of symbols $\sigma \in \Sigma$ in A.

3.12 Definition Let Σ be a ranked alphabet and let $\mathcal{A} = \langle A, \Sigma^A \rangle$ and $\mathcal{B} = \langle B, \Sigma^B \rangle$ be Σ-algebras. Further, let $h : A \to B$ be function that preserves the Σ-algebra structure, i.e. for every $k \geq 0$, every $\sigma \in \Sigma_k$, and every $a_1, \ldots, a_k \in A$,

$$h(\sigma^A(a_1, \ldots, a_k)) \quad = \quad \sigma^B(h(a_1), \ldots, h(a_k)).$$

Then, $h : A \to B$ is called the a Σ*-algebra homomorphism* and will be written as $h : \mathcal{A} \to \mathcal{B}$.

3.13 Remark Similar as for monoids (cf. 2.7), one has to distinct between a Σ-algebra $\mathcal{A}$ and its underlying set A.

3.14 In the previous section, it turned out that the set A^* of all words over an alphabet A is the underlying set of the free monoid generated by A. In the same sense it is the case that for a given set X, the set $T_\Sigma(X)$ is the underlying set of the free Σ-algebra generated by X, namely $F_\Sigma X = \langle T_\Sigma(X), \Sigma^{T_\Sigma(X)}\rangle$ with $\Sigma^{T_\Sigma(X)} = \{\sigma^{T_\Sigma(X)} \mid \sigma \in \Sigma\}$, where for every $k \geq 0$ and $\sigma \in \Sigma_k$, the function $\sigma^{T_\Sigma(X)}$ is defined by

$$\sigma^{T_\Sigma(X)} : T_\Sigma(X)^k \to T_\Sigma(X) : \langle t_1, \ldots, t_k\rangle \mapsto \sigma(t_1, \ldots, t_k).$$

3.15 Proposition *Let Σ be a ranked alphabet. Let X be a set (of variables) and $\mathcal{A} = \langle A, \Sigma^A\rangle$ be a Σ-algebra. Then, for every function $g : X \to A$ there is a unique Σ-algebra homomorphism $\hat{g} : F_\Sigma X \to \mathcal{A}$, such that for every $x \in X$,*

$$\hat{g}(x) = g(x).$$

Proof: I will just sketch the proof here because it can be found in any textbook on universal algebra (e.g. [Wec92, p.18]). The construction of $\hat{g} : F_\Sigma X \to \mathcal{A}$ is similar to the construction in the proof of Proposition 2.17. Let $t \in T_\Sigma(X)$. If $t = x$ for some $x \in X$, then let $\hat{g}(t) = g(x)$. Otherwise $t = \sigma(t_1, \ldots, t_k)$ for some $k \geq 0$, $\sigma \in \Sigma_k$, and $t_1, \ldots, t_k \in T_\Sigma(X)$. Then let $\hat{g}(t) = \sigma^A(\hat{g}(t_1), \ldots, \hat{g}(t_k))$. □

4 Forests and Polyoids

The definition of the set $T_\Sigma(X)$ can be seen as a generalization of the definition of the set A^* in the following sense. If $\Sigma = \Sigma_1 = A$ contains only symbols of rank 1, and $X = X_1 = \{x_1\}$ is a singleton, then the sets $T_\Sigma(X)$ and A^* are isomorphic. This observation is the basis of generalizing the automata theory from words to trees (cf. [TW68]).
However, there is another observation to make. The set $T_\Sigma(X)$ is the underlying set of the free Σ-algebra generated by X, and the set A^* is the underlying set of the free monoid generated by A. This is an asymmetry between words and trees, because the alphabet A does not occur in the definition of a monoid, whereas the ranked alphabet Σ plays a central role in the definition of a Σ-algebra.
In this section, I will resolve this asymmetry by defining the algebraic structure *polyoid* and the notion of *forests* over a ranked alphabet Σ. The set of all forests over a ranked alphabet Σ will be denoted by $\Sigma^{\circledast}$. It will turn out that the set $\Sigma^{\circledast}$ is the underlying set of the free polyoid generated by Σ.

4.1 In Definition 2.3, the concept of an I-indexed set was introduced. In the following definitions, I will use an $\mathbb{N}\times\mathbb{N}$-indexed set A. Following 2.4, I should write $A_{\langle n,m\rangle}$ to denote the set of all those elements of A, which have index $\langle n, m\rangle \in \mathbb{N}\times\mathbb{N}$. However, because of the heavy use of a certain $\mathbb{N}\times\mathbb{N}$-indexed set in the whole thesis and the central role that is played by the natural numbers n and m in the index, I will desist from using the notation from 2.4 in the special case of $\mathbb{N}\times\mathbb{N}$-indexed sets. Instead, I will simplify the notation and write $A[n, m]$ to denote the set of all those elements of A, that have index $\langle n, m\rangle$. In other words, Equation (4) can be restated as

$$A = \bigcup_{n,m \in \mathbb{N}} A[n,m].$$

4.2 Definition Let A be an $\mathbb{N} \times \mathbb{N}$-indexed set. A partial operation $; : A \times A \dashrightarrow A$ is called a *composition operation on* A if for every $a, b \in A$,

- if there are $k, n, m \in \mathbb{N}$ such that $a \in A[k,n]$ and $b \in A[n,m]$, then $;(a,b)$ is defined and $;(a,b) \in A[k,m]$,
- otherwise, $;(a,b)$ is not defined.

Instead of $;(a,b)$, I will use the more readable infix notation $a;b$.

4.3 Definition Let A be an $\mathbb{N} \times \mathbb{N}$-indexed set, and

$$\langle \ldots \rangle : \left(\bigcup_{n,m \in \mathbb{N}} A[n,1]^m \right) \to A$$

be a function. Then, $\langle \ldots \rangle$ is called a *tupling function* if for every $n, m \in \mathbb{N}$ and $a_1, \ldots, a_m \in A[n,1]$,

$$\langle \ldots \rangle (\langle a_1, \ldots, a_m \rangle) \in A[n,m].$$

Instead of $\langle \ldots \rangle (\langle a_1, \ldots, a_m \rangle)$, I will write $\langle\!\langle a_1, \ldots, a_m \rangle\!\rangle$, which is more readable.

4.4 In the following definition, I will generalize the notion of a monoid (cf. Definition 2.6). I have chosen the name *polyoid* following the Greek root $\pi o \lambda \acute{\upsilon}$-, which means *much/many.* — The term *monoid* is derived from the Greek word $\mu \acute{o} \nu o \varsigma$ meaning *single.*[1]

4.5 Definition A *polyoid* $\mathcal{P} = \langle P, ;, \langle \ldots \rangle, \Pi \rangle$ is a quadruple consisting of

- an $\mathbb{N} \times \mathbb{N}$-indexed set P (called the *underlying set* of $\mathcal{P}$), such that for every $n \in \mathbb{N}$, the set $P[n,0]$ is a singleton,
- a composition operation $; : P \times P \dashrightarrow P$,
- a tupling function
$$\langle \ldots \rangle : \left(\bigcup_{n,m \in \mathbb{N}} P[n,1]^m \right) \to P, \text{ and}$$
- an $\mathbb{N}$-indexed set Π of *projections*, where for every $n \in \mathbb{N}$,
$$\Pi_n = \{ \pi_i^n \mid i \in \mathbb{N} \wedge 1 \leq i \leq n \} \subseteq P[n,1],$$

[1] The reader who is familiar with the ancient Greek language knows that $\pi o \lambda \acute{\upsilon} \varsigma$ is the nominative singular form and carries a meaning which is closer to *a lot* than to *many.* To express *many,* one has to use the plural form $\pi o \lambda \lambda o \acute{\iota}$. This means that the name *polloid* would be more appropriate. However, the Latin language contains the verb *polluō* (*to stain*) from which the English word *Pollution* and the Italian word *polluzione* are derived. Further, in the Italian language exists the noun *pollo* (derived from the Latin accusative *pullum*), whose first meaning is *chicken,* but it can also mean *fool* or *jerk.* I hope it is understandable that I preferred the word *polyoid* for phonetical reasons.

such that

(i) for every $n, m \in \mathbb{N}$ and $t \in P[n,m]$,

$$\langle\!\langle \pi_1^n, \ldots, \pi_n^n \rangle\!\rangle; t \;=\; t \;=\; t; \langle\!\langle \pi_1^m, \ldots, \pi_m^m \rangle\!\rangle,$$

(ii) for every $n, m, k, l \in \mathbb{N}$, every $s \in P[n,m]$, every $t \in P[m,k]$, and every $r \in P[k,l]$,

$$(s;t);r \;=\; s;(t;r),$$

(iii) for every $n, m \in \mathbb{N}$, every $t_1, \ldots t_m \in P[n,1]$, and every $i \in \{1, \ldots, m\}$,

$$\langle\!\langle t_1, \ldots, t_m \rangle\!\rangle; \pi_i^m \;=\; t_i,$$

(iv) for every $n, m \in \mathbb{N}$ and $t \in P[n,m]$,

$$\langle\!\langle t; \pi_1^m, \ldots, t; \pi_m^m \rangle\!\rangle \;=\; t.$$

4.6 Remark For every $n \in \mathbb{N}$, the singleton element of the set $P[n,0]$ will be denoted by $\langle\!\langle\rangle\!\rangle_n$.

4.7 Remark Please observe that it follows from the definition that $\Pi_0 = \emptyset$.

4.8 Example A very simple example of a polyoid is $\mathcal{P} = \langle \mathbb{N} \times \mathbb{N}, ;_{\mathcal{P}}, \langle \ldots \rangle_{\mathcal{P}}, \Pi_{\mathcal{P}} \rangle$, where the underlying set $P = \mathbb{N} \times \mathbb{N}$ such that every element $\langle n, m \rangle \in \mathbb{N} \times \mathbb{N}$ has index $\langle n, m \rangle$. In other words $P[n,m] = \{\langle n, m \rangle\}$ is a singleton for every $n, m \in \mathbb{N}$. By Definition 4.5, it is now completely determined, what the operation $;_{\mathcal{P}}$, the tupling function $\langle \ldots \rangle_{\mathcal{P}}$, and the set $\Pi_{\mathcal{P}_1}$ have to look like.

- For any three natural numbers $n, m, k \in \mathbb{N}$,

$$\langle n, m \rangle ;_{\mathcal{P}} \langle m, k \rangle = \langle n, k \rangle.$$

- For any two natural numbers $n, m \in \mathbb{N}$,

$$\langle\!\langle \underbrace{\langle n, 1 \rangle, \ldots, \langle n, 1 \rangle}_{m \text{ pairs}} \rangle\!\rangle_{\mathcal{P}} = \langle n, m \rangle.$$

- The set $\Pi_{\mathcal{P}} = \{\langle n, 1 \rangle \mid n \in \mathbb{N}\}$, where for every $n \in \mathbb{N}$, we have $\Pi_{n\mathcal{P}} = \{\langle n, 1 \rangle\}$ because

$$\pi^n_{1\,\mathcal{P}} \;=\; \ldots \;= \pi^n_{n\,\mathcal{P}} \;=\; \langle n, 1 \rangle.$$

4.9 Example A more complex example is the set $\mathrm{Mat}(R)$ of all matrices in a ring (or semiring) R. If we define for every $n, m \in \mathbb{N}$ the set $\mathrm{Mat}(R)[n,m] = \mathrm{Mat}_{m \times n}(R)$ to be the set of all $m \times n$ matrices, i.e. all matrices

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$

with n rows and m columns, then $\mathcal{P}_{Mat}R = \langle \mathrm{Mat}(R), \cdot, \langle \ldots \rangle, \mathcal{I} \rangle$ is a polyoid, where the operation $\cdot : \mathrm{Mat}(R) \times \mathrm{Mat}(R) \dashrightarrow \mathrm{Mat}(R)$ is the matrix multiplication, and the tupling function

$$\langle \ldots \rangle : \left(\bigcup_{n,m \in \mathbb{N}} \mathrm{Mat}(R)[n,1]^m \right) \to \mathrm{Mat}(R)$$

maps an m-tuple of column vectors (of dimension n) to the $m \times n$ matrix that is obtained by writing these column vectors next to each other. For example, (for $m = 3$, $n = 2$, and $a_{11}, a_{21}, b_{11}, b_{21}, c_{11}, c_{21} \in R$)

$$\left\langle\!\left\langle \begin{pmatrix} a_{11} \\ a_{21} \end{pmatrix}, \begin{pmatrix} b_{11} \\ b_{21} \end{pmatrix}, \begin{pmatrix} c_{11} \\ c_{21} \end{pmatrix} \right\rangle\!\right\rangle = \begin{pmatrix} a_{11} & b_{11} & c_{11} \\ a_{21} & b_{21} & c_{21} \end{pmatrix}.$$

It remains to show what the $\mathbb{N}$-indexed set $\mathcal{I}$ is. In order to do this, let me define for every $n \geq 1$ and $i = 1, \ldots, n$ the column vector

$$I_i^n = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} \in \mathrm{Mat}(R)[n,1] \text{ where for every } j = 1, \ldots, n, \text{ the component } a_j = \begin{cases} 0 & \text{if } j \neq i \\ 1 & \text{if } j = i \end{cases},$$

where $0 \in R$ is the unit element for the addition in the ring R and $1 \in R$ is the unit element for the multiplication in R. Now, for every $n \geq 1$, define the set

$$\mathcal{I}_n = \{I_1^n, \ldots, I_n^n\} \subseteq \mathrm{Mat}(R)[n,1].$$

Let me now illustrate that the conditions (i)-(iv) from Definition 4.5 do indeed hold for this example.

(i) For every $n, m \in \mathbb{N}$ and $M \in \mathrm{Mat}(R)[n,m]$ it holds that

$$\langle\!\langle I_1^n, \ldots, I_n^n \rangle\!\rangle \cdot M \;=\; M \;=\; M \cdot \langle\!\langle I_1^m, \ldots, I_m^m \rangle\!\rangle$$

because for every $n \geq 0$,

$$\langle\!\langle I_1^n, \ldots, I_n^n \rangle\!\rangle = I_{n \times n} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

is the unit matrix.

(ii) The matrix multiplication is associative.

(iii) If for some $n, m \in \mathbb{N}$, a matrix $M \in \mathrm{Mat}(R)[n,m]$ is multiplied with the column vector I_i^m (for some $i = 1, \ldots, m$), then the result is the ith column of M. For example (for $m = 3$, $n = 2$, and $a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23} \in R$)

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} a_{12} \\ a_{22} \end{pmatrix}.$$

(iv) It is obvious that (informally speaking) a matrix M is obtained by writing its columns next to each other.

Remark: For technical reasons, in this example, the existence of an $n \times 0$ matrix (for every $n \in \mathbb{N}$) is assumed.

4.10 Example Another example for a polyoid is the following. Let Q be a set and define for every $n, m \in \mathbb{N}$, the set $\mathrm{Fun}(Q)[n,m] = [Q^m \to Q^n]$ to be the set of all functions $f : Q^n \to Q^m$. Then, $\mathcal{P}_{Fun}Q = \langle \mathrm{Fun}(Q), ;, \langle \ldots \rangle, \Pi \rangle$ is a polyoid, where the concatenation operation $; : F \times F \dashrightarrow F$ realizes the composition of two functions, i.e. for $n, m, k \in \mathbb{N}$ and two functions $f : Q^n \to Q^m$ and $g : Q^m \to Q^k$, let $f; g = g \circ f$. The action of the function $\langle \ldots \rangle$ is defined as follows. For every $n, m \in \mathbb{N}$ and $f_1, \ldots, f_m \in F[n,1]$, define $\langle f_1, \ldots, f_m \rangle(x_1, \ldots, x_n) = \langle f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n) \rangle \in Q^m$ for every $x_1, \ldots, x_n \in Q$. The set Π is the set of projection functions, i.e. for every $n \in \mathbb{N}$ and $i = 1, \ldots, n$, let $\pi_i^n : Q^n \to Q : \langle x_1, \ldots, x_n \rangle \mapsto x_i$.

4.11 Remark The reader who is familiar with category theory may have observed, that a polyoid is just a category (with the natural numbers as objects) that has all finite products and a terminal object. Similarly, a monoid is just a category with only one object. However, I did not want to introduce category theory nor did I want to demand category theory as a prerequisite for the reading of this thesis.
The following proposition restates a well-known property of products in categories.

4.12 Proposition *Let $\mathcal{P} = \langle P, ;, \langle \ldots \rangle, \Pi \rangle$ be a polyoid, and let n, m, k be natural numbers. If $\langle\!\langle s_1, \ldots, s_m \rangle\!\rangle \in P[n,m]$ and $\langle\!\langle t_1, \ldots, t_k \rangle\!\rangle \in P[m,k]$, then*

$$\langle\!\langle s_1, \ldots, s_m \rangle\!\rangle; \langle\!\langle t_1, \ldots, t_k \rangle\!\rangle = \langle\!\langle \langle\!\langle s_1, \ldots, s_m \rangle\!\rangle; t_1, \ldots, \langle\!\langle s_1, \ldots, s_m \rangle\!\rangle; t_m \rangle\!\rangle.$$

Proof: Immediately from (iv), (ii), and (iii) of Definition 4.5.

$$\begin{aligned}
\langle\!\langle s_1, \ldots, s_m \rangle\!\rangle; \langle\!\langle t_1, \ldots, t_k \rangle\!\rangle &= \langle\!\langle (\langle\!\langle s_1, \ldots, s_m \rangle\!\rangle; \langle\!\langle t_1, \ldots, t_k \rangle\!\rangle); \pi_1^k, \ldots, (\langle\!\langle s_1, \ldots, s_m \rangle\!\rangle; \langle\!\langle t_1, \ldots, t_k \rangle\!\rangle); \pi_k^k \rangle\!\rangle \\
&= \langle\!\langle \langle\!\langle s_1, \ldots, s_m \rangle\!\rangle; (\langle\!\langle t_1, \ldots, t_k \rangle\!\rangle; \pi_1^k), \ldots, \langle\!\langle s_1, \ldots, s_m \rangle\!\rangle; (\langle\!\langle t_1, \ldots, t_k \rangle\!\rangle; \pi_k^k) \rangle\!\rangle \\
&= \langle\!\langle \langle\!\langle s_1, \ldots, s_m \rangle\!\rangle; t_1, \ldots, \langle\!\langle s_1, \ldots, s_m \rangle\!\rangle; t_k \rangle\!\rangle
\end{aligned}$$

□

4.13 In Definition 4.5, the requirement (i) says that for every $n \geq 1$, the tuple $\langle\!\langle \pi_1^n, \ldots, \pi_n^n \rangle\!\rangle$ plays a similar role in a polyoid as the unit element in a monoid. For this reason I consider it to be useful to denote this tuple by ε_n. By definition, ε_0 denotes the single element of the set $P[0,0]$. — In category theory, one would write id_n instead of ε_n.

4.14 Remark It follows from Definition 4.5 that for every polyoid $\mathcal{P} = \langle P, ;, \langle \ldots \rangle, \Pi \rangle$ and natural number $n \in \mathbb{N}$, the triple $\langle P[n,n], ;, \varepsilon_n \rangle$ is a monoid. Applied to Example 4.9, this observation restates the well-known fact that for every $n \in \mathbb{N}$, the set of $n \times n$-matrices in a Ring R forms a monoid.

4.15 Definition Let $\mathcal{P} = \langle P, ;_{\mathcal{P}}, \langle \ldots \rangle_{\mathcal{P}}, \Pi_{\mathcal{P}} \rangle$ and $\mathcal{Q} = \langle Q, ;_{\mathcal{Q}}, \langle \ldots \rangle_{\mathcal{Q}}, \Pi_{\mathcal{Q}} \rangle$ be polyoids. Further, let $h : P \to Q$ be a function that preserves the polyoid structure, i.e.

(i) for every $n, m \in \mathbb{N}$, if $p \in P[n,m]$, then $h(p) \in Q[n,m]$,

(ii) for every $n, m, k \in \mathbb{N}$, if $p_1 \in P[n, m]$ and $p_2 \in P[m, k]$, then

$$h(p_1 ;_{\mathcal{P}} p_2) = h(p_1) ;_{\mathcal{Q}} h(p_2),$$

(iii) for every $n, m \in \mathbb{N}$, if $p_1, \ldots, p_m \in P[n, 1]$, then

$$h(\langle\!\langle p_1, \ldots, p_m \rangle\!\rangle_{\mathcal{P}}) = \langle\!\langle h(p_1), \ldots, h(p_m) \rangle\!\rangle_{\mathcal{Q}},$$

(iv) for every $n \geq 1$ and every $i \in \{1, \ldots, n\}$,

$$h(\pi_i^n{}_{\mathcal{P}}) = \pi_i^n{}_{\mathcal{Q}}.$$

Then, $h : P \to Q$ is called a *polyoid homomorphism* and is written as $h : \mathcal{P} \to \mathcal{Q}$.

4.16 Since for every polyoid $\mathcal{P} = \langle P, ;_{\mathcal{P}}, \langle \ldots \rangle_{\mathcal{P}}, \Pi_{\mathcal{P}} \rangle$, the set P is the disjoint union of $N \times N$ nonempty sets, P will never be finite. However, it does make sense to define the notion of a finite polyoid as follows.

4.17 Definition Let $\mathcal{P} = \langle P, ;_{\mathcal{P}}, \langle \ldots \rangle_{\mathcal{P}}, \Pi_{\mathcal{P}} \rangle$ be a polyoid. Then $\mathcal{P}$ is called *finite* if for every $n, m \in \mathbb{N}$, the set $P[n, m]$ is finite.

4.18 Example One example for a finite polyoid is the polyoid $\mathcal{P}_{Fun}Q = \langle \mathrm{Fun}(Q), ;, \langle \ldots \rangle, \Pi \rangle$ from Example 4.10, where Q is a finite set.

4.19 Let me now introduce the notion of forest. All forests over a ranked alphabet Σ form the $\mathbb{N} \times \mathbb{N}$-indexed set $\Sigma^{\circledast}$. Informally speaking, an (n, m)-forest over Σ is an m-tuple of trees over Σ, that can contain at most n variables. An $(n, 1)$-forest consists of only one tree. Here, the word "tree" has only an informal meaning and refers to an object that is slightly different from a tree defined in 3.5. However, it is the case that for every $n, m \in \mathbb{N}$, the set $\Sigma^{\circledast}[n, m]$ of (n, m)-forests is isomorphic to the set $T_\Sigma(X_n)^m$ which has been defined in 3.9.
Furthermore, for every $n \in \mathbb{N}$, the set of $(n, 1)$-forests contains n different *empty trees* $\pi_1^n, \ldots, \pi_n^n$, which are a generalization of the empty word in a free monoid (see 2.6).
An $(n, 0)$-forest is a forest without any trees. Therefore, for every $n \in \mathbb{N}$ there is only one $(n, 0)$-forest, namely the empty tuple of trees. For an $m \in \mathbb{N}$ with $m \neq n$, the unique $(m, 0)$-forest is different from the unique $(n, 0)$-forest.
The following definition spells out the technical details.

4.20 Definition Let Σ be a ranked alphabet. The set $\Sigma^{\circledast}$ of all *forests over* Σ is the smallest $\mathbb{N} \times \mathbb{N}$-indexed set such that

(i) for every $n \geq 0$, the set $\Sigma^{\circledast}[n, 0] = \{\langle\rangle_n\}$ is a singleton,

(ii) for every $n \geq 1$ and every $i \in \{1, \ldots, n\}$, there is a forest $\pi_i^n \in \Sigma^{\circledast}[n, 1]$,

(iii) for every $n, k \geq 0$, every forest $t \in \Sigma^{\circledast}[n, k]$, and every symbol $\sigma \in \Sigma_k$, there is a forest $t\sigma \in \Sigma^{\circledast}[n, 1]$,

(iv) for every $n \geq 0$, $m \geq 2$, and $t_1, \ldots, t_m \in \Sigma^{\circledast}[n, 1]$, there is a forest $\langle t_1, \ldots, t_m \rangle \in \Sigma^{\circledast}[n, m]$.

For every $n, m \geq 0$, an element of $\Sigma^{\circledast}[n, m]$ is called *(n, m)-forest over* Σ.

4.21 Remark For a given ranked alphabet Σ, the $\mathbb{N}\times\mathbb{N}$-indexed set $\Sigma^{\circledast}$ is well-defined because for every $n_1, n_2, m_1, m_2 \in \mathbb{N}$, if $n_1 \neq n_2$ or $m_1 \neq m_2$, then $\Sigma^{\circledast}[n_1, m_1] \cap \Sigma^{\circledast}[n_2, m_2] = \emptyset$.

4.22 For every ranked alphabet Σ and natural number $n \in \mathbb{N}$, define the *unit forest* (or *identity forest*) $\varepsilon_n \in \Sigma^{\circledast}[n, n]$ as in 4.13, i.e.

- $\varepsilon_0 = \langle\rangle_0 \in \Sigma^{\circledast}[0, 0]$, and
- for every $n \geq 1$, let $\varepsilon_n = \langle \pi_1^n, \ldots, \pi_n^n \rangle \in \Sigma^{\circledast}[n, n]$.

4.23 For every $n, m \in \mathbb{N}$, an (n, m)-forest can be visualized as a box with n holes on the left-hand side and m plugs on the right-hand side. As mentioned before, an (n, m)-forest consists of m trees. Each of the m trees is connected to one of the m plugs of the box. For every $n \in \mathbb{N}$, the forest $\langle\rangle_n$ has n holes, no plugs and does not contain anything. For that reason, I will call it *empty forest*. Figure 8 shows the empty forest for $n = 3$. Figure 8 also shows the forests π_1^3, π_2^3, π_3^3. Since they are (3,1)-forests, they can be considered as trees. But they do not contain any symbols from the alphabet Σ. For that reason, I will call them *empty trees*. The last forest in Figure 8 is the unit forest for $n = 3$.

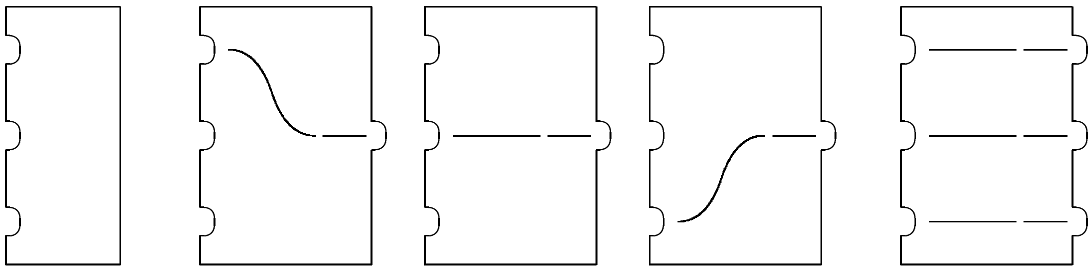

Figure 8: The empty forests $\langle\rangle_3$, the empty trees π_1^3, π_2^3, π_3^3, and the unit forest $\varepsilon_3 = \langle \pi_1^3, \pi_2^3, \pi_3^3 \rangle$.

4.24 Example In order to see more complex forests, let us consider again $\Sigma = \Sigma_0 \cup \Sigma_2$ with $\Sigma_0 = \{\alpha\}$ and $\Sigma_2 = \{\sigma\}$. Then, we have $\pi_1^2, \pi_2^2 \in \Sigma^{\circledast}[2, 1]$ by definition. Other examples are e.g.

$$\begin{aligned}
\langle \pi_1^2, \pi_2^2 \rangle = \varepsilon_2 &\in \Sigma^{\circledast}[2, 2], \\
\langle\rangle_2 \alpha &\in \Sigma^{\circledast}[2, 1], \\
\langle\rangle_0 \alpha = \varepsilon_0 \alpha &\in \Sigma^{\circledast}[0, 1], \\
\langle \langle\rangle_2 \alpha, \pi_1^2 \rangle \sigma &\in \Sigma^{\circledast}[2, 1], \\
\langle \langle \pi_1^2, \pi_2^2 \rangle \sigma, \pi_1^2, \pi_2^2 \rangle = \langle \varepsilon_2 \sigma, \pi_1^2, \pi_2^2 \rangle &\in \Sigma^{\circledast}[2, 3], \\
\langle \langle\rangle_2 \alpha, \langle \langle\rangle_2 \alpha, \pi_1^2 \rangle \sigma, \pi_1^2 \rangle &\in \Sigma^{\circledast}[2, 3], \\
\langle \langle \pi_1^3, \langle\rangle_3 \alpha \rangle \sigma, \pi_2^3 \rangle &\in \Sigma^{\circledast}[3, 2], \\
\langle \langle \pi_2^2, \pi_2^2 \rangle \sigma, \pi_1^2 \rangle \sigma &\in \Sigma^{\circledast}[2, 1].
\end{aligned}$$

The last three of them are shown in Figure 9.

4.25 Remark In order to avoid unnecessary case analyses, a forest $t \in \Sigma^{\circledast}[n, 1]$ will sometimes be denoted by $\langle t \rangle$.

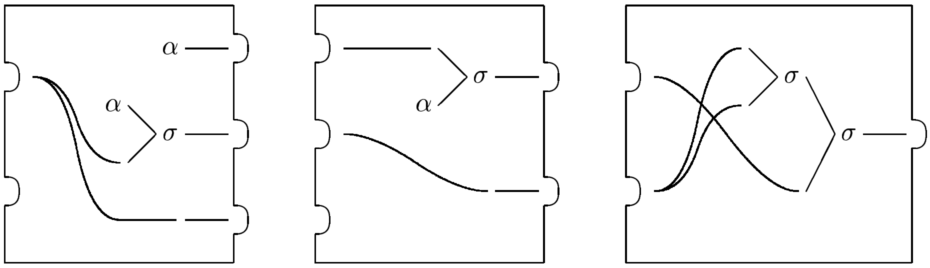

Figure 9: The forests $\langle\langle\rangle_2\alpha, \langle\langle\rangle_2\alpha, \pi_1^2\rangle\sigma, \pi_1^2\rangle$, $\langle\langle\pi_1^3, \langle\rangle_3\alpha\rangle\sigma, \pi_2^3\rangle$, and $\langle\langle\pi_2^2, \pi_2^2\rangle\sigma, \pi_1^2\rangle\sigma$.

4.26 Lemma *For every ranked alphabet Σ and natural numbers $n, m \in \mathbb{N}$, the sets $\Sigma^{\circledast}[n,1]^m$ and $\Sigma^{\circledast}[n,m]$ are isomorphic.*

Proof: Immediate from Definition 4.20. □

4.27 Proposition *For a given ranked alphabet Σ and $n \in \mathbb{N}$, the function*

$$\varphi : \Sigma^{\circledast}[n,1] \quad \to \quad T_\Sigma(X_n),$$

where for every $i = 1, \ldots, n$, every $k \geq 0$, every $\sigma \in \Sigma_k$, and every $t_1, \ldots, t_k \in \Sigma^{\circledast}[n,1]$,

$$\begin{aligned} \varphi(\pi_i^n) &= x_i, \quad \textit{and} \\ \varphi(\langle t_1, \ldots, t_k\rangle\sigma) &= \sigma(\varphi(t_1), \ldots, \varphi(t_k)), \end{aligned}$$

is a bijection.

Proof: Immediate from Definitions 3.5 and 4.20. □

4.28 Remark The bijection $\varphi : \Sigma^{\circledast}[n,1] \to T_\Sigma(X_n)$ is a Σ-algebra isomorphism.

4.29 Corollary *For every ranked alphabet Σ and natural numbers $n, m \in \mathbb{N}$, the sets $T_\Sigma(X_n)^m$ and $\Sigma^{\circledast}[n,m]$ are isomorphic.*

Proof: Immediate from Lemma 4.26 and Proposition 4.27. □

4.30 Although the sets $\Sigma^{\circledast}[n,1]$ and $T_\Sigma(X_n)$ are isomorphic, there is a subtle difference that should not be neglected. According to Definition 3.5, it is the case that $T_\Sigma(X_{n_1}) \subset T_\Sigma(X_{n_2})$ if $n_1 < n_2$. But it will never be the case that $\Sigma^{\circledast}[n_1,1] \subset \Sigma^{\circledast}[n_2,1]$ if $n_1 \neq n_2$. Even more, it is the case that $\Sigma^{\circledast}[n_1,1] \cap \Sigma^{\circledast}[n_2,1] = \emptyset$ whenever $n_1 \neq n_2$.

4.31 Remark From Definition 4.20, it follows that is possible to prove statements on forests $t \in \Sigma^{\circledast}$ using structural induction by taking care of the following four cases.

(i) $t = \langle\rangle_n$ (for some $n \in \mathbb{N}$),

(ii) $t = \pi_i^n$ (for some $n \geq 1$ and $i = 1, \ldots, n$),

(iii) $t = t'\sigma$ (for some $n, k \geq 0$, $t' \in \Sigma^{\circledast}[n,k]$ and $\sigma \in \Sigma_k$), and

(iv) $t = \langle t_1, \ldots, t_m\rangle$ (for some $n \geq 0$, $m \geq 2$ and $t_1, \ldots, t_m \in \Sigma^{\circledast}[n,1]$).

Similarly, definitions can be done inductively on the structure of $t \in \Sigma^{\circledast}$.

4.32 For a given ranked alphabet Σ, define the *forest concatenation operation*

$$; : \Sigma^{\circledast} \times \Sigma^{\circledast} \dashrightarrow \Sigma^{\circledast}$$

inductively as follows.

(i) For every $n, m \in \mathbb{N}$ and $t \in \Sigma^{\circledast}[n,m]$,

$$t; \langle\rangle_m \;=\; \langle\rangle_n,$$

(ii) for every $n, m \in \mathbb{N}$ (with $m \geq 1$), every $t_1, \ldots, t_m \in \Sigma^{\circledast}[n,1]$, and every $i \in \{1, \ldots, m\}$,

$$\langle t_1, \ldots, t_m\rangle; \pi_i^m \;=\; t_i,$$

(iii) for every $n, m, k \in \mathbb{N}$, every $s \in \Sigma^{\circledast}[n,m]$, and every $t \in \Sigma^{\circledast}[m,k]$, and every $\sigma \in \Sigma_k$,

$$s; (t\sigma) \;=\; (s;t)\sigma,$$

(iv) for every $n, m, k \in \mathbb{N}$ (with $k \geq 2$), every $s \in \Sigma^{\circledast}[n,m]$, and every $t_1, \ldots, t_k \in \Sigma^{\circledast}[m,1]$,

$$s; \langle t_1, \ldots, t_k\rangle \;=\; \langle s;t_1, \ldots, s;t_k\rangle.$$

4.33 Lemma *For every ranked alphabet Σ, the forest concatenation operation $; : \Sigma^{\circledast} \times \Sigma^{\circledast} \dashrightarrow \Sigma^{\circledast}$ is associative, i.e. for every $n, m, k, l \in \mathbb{N}$, every $s \in \Sigma^{\circledast}[n,m]$, every $t \in \Sigma^{\circledast}[m,k]$, and every $r \in \Sigma^{\circledast}[k,l]$,*

$$(s;t);r \;=\; s;(t;r).$$

Proof: By case analysis on l and induction on r.

- $\underline{l = 0.}$ Then $r = \langle\rangle_k$. Therefore

$$(s;t);r = (s;t);\langle\rangle_k = \langle\rangle_n = s;\langle\rangle_m = s;(t;\langle\rangle_k) = s;(t;r).$$

- $\underline{l = 1.}$ Then there are two cases:
  - $\underline{r = \pi_i^k}$ for some $i = 1, \ldots, k$.
    Let $t = \langle t_1, \ldots, t_k\rangle$ where $t_1, \ldots, t_k \in \Sigma^{\circledast}[m,1]$. Then

$$(s;t);r = (s;\langle t_1, \ldots, t_k\rangle);\pi_i^k = \langle s;t_1, \ldots, s;t_k\rangle;\pi_i^k = s;t_i = s;(t;\pi_i^k) = s;(t;r).$$

  - $\underline{r = r'\sigma}$ for some $k' \in \mathbb{N}$ and $\sigma \in \Sigma_{k'}$ and $r' \in \Sigma^{\circledast}[k,k']$. Then

$$\begin{aligned}
(s;t);r &= (s;t);r'\sigma \\
&= ((s;t);r')\sigma \quad \text{(by 4.32.(iii))} \\
&= (s;(t;r'))\sigma \quad \text{(by induction hypothesis)} \\
&= s;(t;r')\sigma \quad \text{(by 4.32.(iii))} \\
&= s;(t;r'\sigma) \quad \text{(by 4.32.(iii))} \\
&= s;(t;r).
\end{aligned}$$

- $\underline{l \geq 2.}$ Then $r = \langle r_1, \ldots, r_l \rangle$ where $r_1, \ldots, r_l \in \Sigma^{\circledast}[k, 1]$. Therefore

$$\begin{aligned}
(s;t);r &= (s;t);\langle r_1, \ldots, r_l \rangle \\
&= \langle (s;t);r_1, \ldots, (s;t);r_l \rangle \quad \text{(by 4.32.(iv))} \\
&= \langle s;(t;r_1), \ldots, s;(t;r_l) \rangle \quad \text{(by induction hypothesis)} \\
&= s;\langle t;r_1, \ldots, t;r_l \rangle) \quad \text{(by 4.32.(iv))} \\
&= s;(t;\langle r_1, \ldots, r_l \rangle) \quad \text{(by 4.32.(iv))} \\
&= s;(t;r).
\end{aligned}$$

□

4.34 Example Consider again $\Sigma = \Sigma_0 \cup \Sigma_2$ with $\Sigma_0 = \{\alpha\}$ and $\Sigma_2 = \{\sigma\}$. Let

$$s = \langle \langle \pi_1^3, \langle\rangle_3 \alpha \rangle \sigma, \pi_2^3 \rangle \in \Sigma^{\circledast}[3, 2] \quad \text{and} \quad t = \langle \langle \pi_2^2, \pi_2^2 \rangle \sigma, \pi_1^2 \rangle \sigma \in \Sigma^{\circledast}[2, 1].$$

Both forest are shown in Figure 9. Their concatenation can formally be computed as follows.

$$\begin{aligned}
s;t &= \langle \langle \pi_1^3, \langle\rangle_3 \alpha \rangle \sigma, \pi_2^3 \rangle ; \langle \langle \pi_2^2, \pi_2^2 \rangle \sigma, \pi_1^2 \rangle \sigma \\
&= (\langle \langle \pi_1^3, \langle\rangle_3 \alpha \rangle \sigma, \pi_2^3 \rangle ; \langle \langle \pi_2^2, \pi_2^2 \rangle \sigma, \pi_1^2 \rangle) \sigma \\
&= \langle \langle \langle \pi_1^3, \langle\rangle_3 \alpha \rangle \sigma, \pi_2^3 \rangle ; \langle \pi_2^2, \pi_2^2 \rangle \sigma, \langle \langle \pi_1^3, \langle\rangle_3 \alpha \rangle \sigma, \pi_2^3 \rangle ; \pi_1^2 \rangle \sigma \\
&= \langle (\langle \langle \pi_1^3, \langle\rangle_3 \alpha \rangle \sigma, \pi_2^3 \rangle ; \langle \pi_2^2, \pi_2^2 \rangle) \sigma, \langle \pi_1^3, \langle\rangle_3 \alpha \rangle \sigma \rangle \sigma \\
&= \langle \langle \langle \langle \pi_1^3, \langle\rangle_3 \alpha \rangle \sigma, \pi_2^3 \rangle ; \pi_2^2, \langle \langle \pi_1^3, \langle\rangle_3 \alpha \rangle \sigma, \pi_2^3 \rangle ; \pi_2^2 \rangle \sigma, \langle \pi_1^3, \langle\rangle_3 \alpha \rangle \sigma \rangle \sigma \\
&= \langle \langle \pi_2^3, \pi_2^3 \rangle \sigma, \langle \pi_1^3, \langle\rangle_3 \alpha \rangle \sigma \rangle \sigma \in \Sigma^{\circledast}[3, 1]
\end{aligned}$$

Figure 10 might help to understand the mechanism.

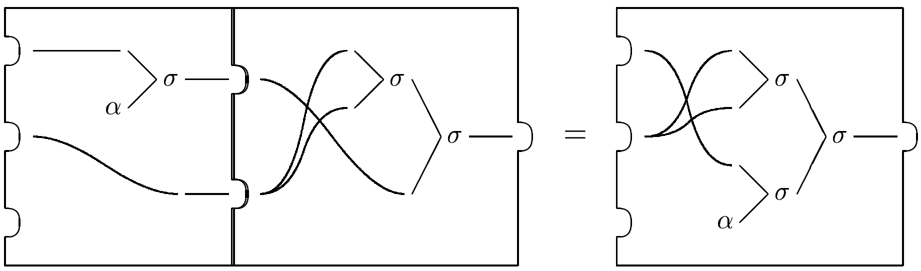

Figure 10: The concatenation $s;t$.

4.35 Remark It should be obvious now that the forest concatenation operation simply realizes a term substitution. This can be spelled out formally as follows. Let Σ be a ranked alphabet and $n \geq 0$, $m \geq 1$ be natural numbers. Further, let $s_1, \ldots, s_m \in \Sigma^{\circledast}[n, 1]$ and $t \in \Sigma^{\circledast}[m, 1]$. Then, taking the bijection φ from Proposition 4.27,

$$\varphi(\langle s_1, \ldots, s_m \rangle; t) = \varphi(t)[x_1 \leftarrow \varphi(s_1), \ldots, x_m \leftarrow \varphi(s_m)].$$

In other words, the Σ-term $\varphi(\langle s_1, \ldots, s_m \rangle; t) \in T_\Sigma(X_n)$ is obtained by replacing for every $i = 1, \ldots, m$, each occurrence of $x_i \in X_m$ in the Σ-tree $\varphi(t) \in T_\Sigma(X_m)$ by the Σ-tree $\varphi(s_i) \in T_\Sigma(X_n)$.

4.36 Let Σ be a ranked alphabet. Then define the *forest tupling function* $\langle \ldots \rangle$ on $\Sigma^{\circledast}$ by

$$\langle \ldots \rangle : \left(\bigcup_{n,m \in \mathbb{N}} \Sigma^{\circledast}[n,1]^m \right) \to \Sigma^{\circledast} : \langle t_1, \ldots, t_m \rangle \mapsto \langle t_1, \ldots, t_m \rangle.$$

In other words, for every $t_1, \ldots, t_m \in \Sigma^{\circledast}[n,1]$, the tuple $\langle t_1, \ldots, t_m \rangle \in \Sigma^{\circledast}[n,1]^m$ (i.e. the element of the cartesian product as defined in 2.1) is mapped to the forest $\langle t_1, \ldots, t_m \rangle \in \Sigma^{\circledast}[n,m]$ (as defined in (iii) of Definition 4.20). This can also be written as

$$\langle\!\langle t_1, \ldots, t_m \rangle\!\rangle \;=\; \langle t_1, \ldots, t_m \rangle.$$

4.37 It follows now that the $\mathbb{N} \times \mathbb{N}$-indexed set $\Sigma^{\circledast}$ together with the forest concatenation operation, the forest tupling function, and the empty trees as projections does form a polyoid.

4.38 Proposition *Let Σ be a ranked alphabet. Then the tuple $F_{\mathrm{Pol}}\Sigma = \langle \Sigma^{\circledast}, ;, \langle \ldots \rangle, \Pi \rangle$ is a polyoid, where the underlying set $\Sigma^{\circledast}$ is the set of all forests over Σ, the operation ; is the forest concatenation operation, the function $\langle \ldots \rangle$ is the forest tupling function, and the set Π of projections is the $\mathbb{N}$-indexed set of empty trees, namely, $\Pi_n = \{\pi^n_1, \ldots, \pi^n_n\} \subseteq \Sigma^{\circledast}[n,1]$ for every $n \in \mathbb{N}$. The polyoid $F_{\mathrm{Pol}}\Sigma$ is called the* free polyoid generated by *Σ.*

Proof: It remains to show that conditions (i)-(iv) of Definition 4.5 hold for $F_{\mathrm{Pol}}\Sigma$.

(i) The first equation follows from the fact that $\langle \pi^n_1, \ldots, \pi^n_n \rangle ; \pi^n_i = \pi^n_i$ for every $n \geq 1$ and $i = 1, \ldots, n$, and a similar inductions as in the proof for Lemma 4.33. The second equation follows immediately from (iv) and (ii) of 4.32.

(ii) Lemma 4.33.

(iii) Immediate from 4.32.(ii).

(iv) Immediate from Definition 4.20, from 4.32.(ii), and from 4.36.

□

4.39 The name "free polyoid" will be justified by a similar property as stated in Propositions 2.17 and 3.15. However, in order to extend (for a given ranked alphabet Σ and a polyoid $\mathcal{P} = \langle P, ;_{\mathcal{P}}, \langle \ldots \rangle_{\mathcal{P}}, \Pi_{\mathcal{P}} \rangle$) a function $g : \Sigma \to P$ to a polyoid homomorphism $\hat{g} : F_{\mathrm{Pol}}\Sigma \to \mathcal{P}$, we have to solve a technical problem because g (going from an $\mathbb{N}$-indexed set to an $\mathbb{N} \times \mathbb{N}$-indexed set) cannot be any function. We can allow only those functions that map a symbol $\sigma \in \Sigma_k$ (for some $k \in \mathbb{N}$) to an element of $P[k,1]$.

4.40 Definition Let A be an $\mathbb{N}$-indexed set and B be an $\mathbb{N} \times \mathbb{N}$-indexed set. A function $f : A \to B$ is called *index preserving* if for every $k \in \mathbb{N}$ and every $a \in A_k$, it holds that $f(a) \in B[k,1]$.

4.41 Proposition *Let Σ be a ranked alphabet and $\mathcal{P} = \langle P, ;_{\mathcal{P}}, \langle \ldots \rangle_{\mathcal{P}}, \Pi_{\mathcal{P}} \rangle$ be a polyoid. Then, for every index preserving function $g : \Sigma \to P$, there is a unique polyoid homomorphism $\hat{g} : F_{\mathrm{Pol}}\Sigma \to \mathcal{P}$ such that for every $k \in \mathbb{N}$ and $\sigma \in \Sigma_k$,*

$$\hat{g}(\varepsilon_k \sigma) \;=\; g(\sigma). \tag{6}$$

Proof: The proof is essentially the same as the proofs for Propositions 2.17 and 3.15. Construct $\hat{g} : F_{\mathrm{Pol}}\Sigma \to \mathcal{P}$ inductively as follows.

(i) For every $n \geq 0$, let $\hat{g}(\langle\rangle_n) = \langle\!\langle\rangle\!\rangle_{n\mathcal{P}}$.

(ii) For every $n \geq 1$ and $i = 1, \ldots, n$, let $\hat{g}(\pi_i^n) = \pi_i^n{}_{\mathcal{P}}$.

(iii) For every $n, k \geq 0$ and $\sigma \in \Sigma_k$ and $t \in \Sigma^{\circledast}[n,k]$, let $\hat{g}(t\sigma) = \hat{g}(t);_{\mathcal{P}} g(\sigma)$.

(iv) For every $n \geq 0$, $m \geq 2$ and $t = \langle t_1, \ldots, t_m\rangle \in \Sigma^{\circledast}[n,m]$, let $\hat{g}(t) = \langle\!\langle \hat{g}(t_1), \ldots, \hat{g}(t_m)\rangle\!\rangle_{\mathcal{P}}$.

In order to show that $\hat{g}$ is indeed a polyoid homomorphism, it remains to show that for every $m, n, k \in \mathbb{N}$ and $s \in \Sigma^{\circledast}[n,m]$ and $t \in \Sigma^{\circledast}[m,k]$, we have

$$\hat{g}(s;t) = \hat{g}(s);_{\mathcal{P}} \hat{g}(t).$$

But this follows by an easy induction on the structure of t. The only nontrivial case is $t = t'\sigma$ (i.e. $k = 1$) for some $k' \geq 0$, some $t' \in \Sigma^{\circledast}[m,k']$ and $\sigma \in \Sigma_{k'}$. Then

$$\begin{array}{rcl}
\hat{g}(s;t) & = & \hat{g}(s;t'\sigma) \;=\; \hat{g}((s;t')\sigma) \;\; \text{(by 4.32.(iii))} \\
& = & \hat{g}(s;t');_{\mathcal{P}} g(\sigma) \;\; \text{(by (iii))} \\
& = & (\hat{g}(s);_{\mathcal{P}} \hat{g}(t'));_{\mathcal{P}} g(\sigma) \;\; \text{(by induction hypothesis)} \\
& = & \hat{g}(s);_{\mathcal{P}} (\hat{g}(t');_{\mathcal{P}} g(\sigma)) \;\; \text{(by 4.5.(ii))} \\
& = & \hat{g}(s);_{\mathcal{P}} \hat{g}(t'\sigma) \;\; \text{(by (iii))} \\
& = & \hat{g}(s);_{\mathcal{P}} \hat{g}(t)
\end{array}$$

Let me now show that the constructed $\hat{g}$ satisfies equation (6).

$$\hat{g}(\varepsilon_k\sigma) = \hat{g}(\varepsilon_k);_{\mathcal{P}} g(\sigma) = \hat{g}(\langle\pi_1^k, \ldots, \pi_k^k\rangle);_{\mathcal{P}} g(\sigma) = \langle\!\langle \pi_1^k{}_{\mathcal{P}}, \ldots, \pi_k^k{}_{\mathcal{P}}\rangle\!\rangle;_{\mathcal{P}} g(\sigma) = g(\sigma).$$

In order to complete the proof, it remains to show that $\hat{g}$ is unique. Assume, there is a polyoid homomorphism $\hat{g}' : F_{\mathrm{Pol}}\Sigma \to \mathcal{P}$ with $\hat{g}'(\varepsilon_k\sigma) = g(\sigma)$ for every $k \geq 0$ and $\sigma \in \Sigma_k$. Then it follows by an easy induction that $\hat{g}' = \hat{g}$:

(i) For every $n \geq 0$, we have $\hat{g}'(\langle\rangle_n) = \langle\!\langle\rangle\!\rangle_n = \hat{g}(\langle\rangle_n)$ because $\hat{g}'$ is a homomorphism,

(ii) for every $n \geq 1$ and $i = 1, \ldots, n$, we have $\hat{g}'(\pi_i^n) = \pi_i^n{}_{\mathcal{P}} = \hat{g}(\pi_i^n)$ because $\hat{g}'$ is a homomorphism,

(iii) for every $n, k \geq 0$ and $\sigma \in \Sigma_k$ and $t \in \Sigma^{\circledast}[n,k]$, we have $\hat{g}'(t\sigma) = \hat{g}'(t;\varepsilon_k\sigma) = \hat{g}'(t);_{\mathcal{P}} \hat{g}'(\varepsilon_k\sigma) = \hat{g}(t);_{\mathcal{P}} g(\sigma) = \hat{g}(t\sigma)$ by induction hypothesis and because $\hat{g}'$ is a homomorphism, and

(iv) for every $n \geq 0$, $m \geq 2$ and $t = \langle t_1, \ldots, t_m\rangle \in \Sigma^{\circledast}[n,m]$, we have $\hat{g}'(t) = \langle\!\langle \hat{g}'(t_1), \ldots, \hat{g}'(t_m)\rangle\!\rangle_{\mathcal{P}} = \langle\!\langle \hat{g}(t_1), \ldots, \hat{g}(t_m)\rangle\!\rangle_{\mathcal{P}} = \hat{g}(t)$ by induction hypothesis and because $\hat{g}'$ is a homomorphism.

□

4.42 Remark The construction of Proposition 4.41 might seem more complicated than the constructions of Propositions 2.17 and 3.15. However, from a categorical point of view all three constructions are instances of the same general concept. Let **Set** denote the category of sets and functions, **Mon** the category of monoids and monoid homomorphisms, **Σ-Alg** the category of Σ-algebras and Σ-algebra homomorphisms (for a given ranked alphabet Σ), and **Pol** the category of polyoids and polyoid homomorphisms.

Further, let U_{Mon} : **Mon** → **Set** be the functor that takes a monoid $\mathcal{M}$ to its underlying set M. If F_{Mon} : **Set** → **Mon** is the functor that takes a set A to the free monoid $F_{\mathrm{Mon}}A$, then $F_{\mathrm{Mon}} \dashv U_{\mathrm{Mon}}$ is a pair of adjoint functors.

Similarly, let (for a given ranked alphabet Σ) U_Σ : **Σ-Alg** → **Set** be the functor that takes a Σ-algebra $\mathcal{A}$ to its underlying set A. If F_Σ : **Set** → **Σ-Alg** is the functor that takes a set X to

the free Σ-algebra $F_\Sigma X$, then $F_\Sigma \dashv U_\Sigma$ is another pair of adjoint functors.
Let $\mathbf{Set}/\mathbb{N}$ denote the slice category (cf. [BW99, p.18]) of $\mathbb{N}$-indexed sets, and let $U_{\mathrm{Pol}} : \mathbf{Pol} \to \mathbf{Set}/\mathbb{N}$ be the functor that takes a polyoid $\mathcal{P}$ to the $\mathbb{N}$-indexed set $\{p \in P[n,1] \mid n \in \mathbb{N}\}$ (where P is the underlying set of $\mathcal{P}$). If $F_{\mathrm{Pol}} : \mathbf{Set}/\mathbb{N} \to \mathbf{Pol}$ is the functor that takes an $\mathbb{N}$-indexed set Σ to the free polyoid $F_{\mathrm{Pol}}\Sigma$, then $F_{\mathrm{Pol}} \dashv U_{\mathrm{Pol}}$ is also a pair of adjoint functors.

4.43 Definition Let Σ be a ranked alphabet. Then, define a function

$$\mathit{size}_\Sigma : \Sigma^{\circledast} \to \mathbb{N}$$

inductively as follows.

(i) For every $n \geq 0$,

$$\mathit{size}_\Sigma(\langle\rangle_n) = 0.$$

(ii) For every $n \geq 1$ and $i = 1, \ldots, n$,

$$\mathit{size}_\Sigma(\pi_i^n) = 0.$$

(iii) For every $n, k \geq 0$, every $t \in \Sigma^{\circledast}[n,k]$, and every $\sigma \in \Sigma_k$,

$$\mathit{size}_\Sigma(t\sigma) = \mathit{size}_\Sigma(t) + 1.$$

(iv) For every $n \geq 0, m \geq 2$, and $t_1, \ldots, t_m \in \Sigma^{\circledast}[n,1]$,

$$\mathit{size}_\Sigma(\langle t_1, \ldots, t_m\rangle) = \mathit{size}_\Sigma(t_1) + \ldots + \mathit{size}_\Sigma(t_m).$$

4.44 Remark When I use the function *size*, I will omit the index Σ where no ambiguity arises.

4.45 Remark Note the difference to the *size* function for trees that can be found in the literature (cf. [FV98, pp.58]), where the size of a tree containing only a variable is one, i.e. $\mathit{size}(x_i) = 1$, whereas in my definition $\mathit{size}(\pi_i^n) = 0$. I justify my definition with the observation that in the free monoid the length of the empty word is zero and not one.

4.46 Remark Another observation that should be mentioned is that for two forests $s \in \Sigma^{\circledast}[n,m]$ and $t \in \Sigma^{\circledast}[m,k]$, the value of $\mathit{size}(s;t)$ is in general not equal to $\mathit{size}(s) + \mathit{size}(t)$. Consider for example $\Sigma = \Sigma_0 \cup \Sigma_2$ with $\Sigma_0 = \{\alpha\}$ and $\Sigma_2 = \{\sigma\}$. Let $s = \langle\rangle_2\alpha \in \Sigma^{\circledast}[2,1]$ and $t = \langle\pi_1^1, \pi_1^1\rangle\sigma \in \Sigma^{\circledast}[1,1]$. Then $\mathit{size}(s) = 1$ and $\mathit{size}(t) = 1$. Hence $\mathit{size}(s) + \mathit{size}(t) = 2$. But $s;t = \langle\langle\rangle_2\alpha, \langle\rangle_2\alpha\rangle\sigma \in \Sigma^{\circledast}[2,1]$ and $\mathit{size}(s;t) = 3$.

4.47 Definition For a given ranked alphabet Σ, and two natural numbers n and m, a subset $L \subseteq \Sigma^{\circledast}[n,m]$ will be called a *(n,m)-forest language over Σ*.

5 Recognizable Forest Languages

In this section, I will define recognizable forest languages by introducing finite-state forest automata which accept those languages. A finite-state forest automaton is a simple extension of a finite-state tree automaton as it can be found in the literature (e.g. [GS97]). The main difference is, that finite-state forest automata have initial as well as final states, whereas finite-state tree automata do only have final states (or, in the top-down case, only initial states). Observe, that usual finite-state automata that operate on words do also have initial as well as final states. Perhaps there is nothing deep behind this observation but in some sense finite-state forest automata regain a symmetry which was lost for finite-state tree automata. In developing the theory of forest automata, I will stay as close as possible to the theory of finite-state automata for word languages (cf. [Per90]). The main difficulty is that there is no longer a single state transition relation (or state transition function), but a family of such relations (or functions), one for each rank that occurs in the ranked alphabet Σ.

5.1 Definition Let n and m be natural numbers. A *(nondeterministic) finite-state* (n, m) *forest automaton* (*(n, m)-fsfa*) is a tuple $\mathfrak{A} = \langle Q, \Sigma, I, F, E \rangle$, where Q is a finite set of *states*, Σ is a ranked alphabet, $I = I_1 \times \ldots \times I_n$ (with $I_1, \ldots, I_n \subseteq Q$) is the set of *initial state tuples*, $F = F_1 \times \ldots \times F_m$ (with $F_1, \ldots, F_m \subseteq Q$) is the set of *final state tuples*, and

$$E = \{E_k \mid k \geq 0\}, \quad \text{where} \quad E_k \subseteq Q^k \times \Sigma_k \times Q \quad (\text{for every } k \geq 0)$$

is the set of *state transition relations*.

5.2 Remark It might be helpful to notice that for almost all $k \in \mathbb{N}$ the relation E_k is empty, because Σ is a finite set. However, the following elaboration will not suffer from this fact.

5.3 For a given fsfa $\mathfrak{A} = \langle Q, \Sigma, I, F, E \rangle$ with $E = \{E_k \mid k \geq 0\}$, I will now define for every $k, l \geq 0$ a relation $E^{\mathfrak{A}}_{k,l} \subseteq (\mathfrak{P}(Q))^k \times \Sigma^{\circledast}[k, l] \times Q^l$. The definition is done inductively on the structure of the elements of $\Sigma^{\circledast}[k, l]$.

(i) For every $k \geq 0$ and $Q_1, \ldots, Q_k \subseteq Q$,

$$(\langle Q_1, \ldots, Q_k \rangle, \langle\rangle_k, \langle\rangle) \in E^{\mathfrak{A}}_{k,0},$$

where $\langle\rangle_k \in \Sigma^{\circledast}[k, 0]$ is the empty forest (cf. Definition 4.20) and $\langle\rangle \in Q^0$ is the empty tuple of states (cf. 2.1),

(ii) for every $k \geq 0$, every $Q_1, \ldots, Q_k \subseteq Q$, every $q \in Q$, and every $i = 1, \ldots, k$, if $q \in Q_i$, then

$$(\langle Q_1, \ldots, Q_k \rangle, \pi^k_i, q) \in E^{\mathfrak{A}}_{k,1},$$

(iii) for every $k \geq 0$, every $Q_1, \ldots, Q_k \subseteq Q$, and every $q \in Q$, if there are a natural number $k' \geq 0$, a forest $t \in \Sigma^{\circledast}[k, k']$, a symbol $\sigma \in \Sigma_{k'}$, and states $p_1, \ldots, p_{k'} \in Q$ such that

- $(\langle Q_1, \ldots, Q_k \rangle, t, \langle p_1, \ldots, p_{k'} \rangle) \in E^{\mathfrak{A}}_{k,k'}$ and
- $(\langle p_1, \ldots, p_{k'} \rangle, \sigma, q) \in E_{k'}$,

then $(\langle Q_1, \ldots, Q_k \rangle, t\sigma, q) \in E^{\mathfrak{A}}_{k,1}$,

(iv) for every $k \geq 0$ and $l \geq 2$, every $Q_1, \ldots, Q_k \subseteq Q$, every $p_1, \ldots, p_l \in Q$, and every $t_1, \ldots, t_l \in \Sigma^{\circledast}[k, 1]$, if

$$\begin{aligned} (\langle Q_1, \ldots, Q_k \rangle, t_1, p_1) &\in E^{\mathfrak{A}}_{k,1}, \\ &\vdots \\ (\langle Q_1, \ldots, Q_k \rangle, t_l, p_l) &\in E^{\mathfrak{A}}_{k,1}, \end{aligned}$$

then $(\langle Q_1, \ldots, Q_k \rangle, \langle t_1, \ldots, t_l \rangle, \langle p_1, \ldots, p_l \rangle) \in E^{\mathfrak{A}}_{k,l}$.

For every $k, l \in \mathbb{N}$, the relation $E_{k,l}^{\mathfrak{A}}$ does not contain any other elements.

5.4 Definition Let $n, m \in \mathbb{N}$ and let $\mathfrak{A} = \langle Q, \Sigma, I, F, E\rangle$ be an (n,m)-fsfa. Then

$$L(\mathfrak{A}) \quad = \quad \{t \in \Sigma^{\circledast}[n,m] \mid \exists \langle p_1, \ldots, p_m\rangle \in F \;.\; (\langle I_1, \ldots, I_n\rangle, t, \langle p_1, \ldots, p_m\rangle) \in E_{n,m}^{\mathfrak{A}}\}$$

is the language *accepted* (or *recognized*) by $\mathfrak{A}$.

5.5 Example Consider $\Sigma = \Sigma_0 \cup \Sigma_2$ with $\Sigma_0 = \{\alpha\}$ and $\Sigma_2 = \{\sigma\}$. Define a $(3,1)$-fsfa $\mathfrak{A} = \langle Q, \Sigma, I, F, E\rangle$ as follows.

$$\begin{array}{rcl} Q & = & \{e, o\} \\ I & = & \{e\} \times \{e\} \times \{e\} \;=\; \{\langle e,e,e\rangle\} \\ F & = & \{e\} \\ E_0 & = & \{(\langle\rangle, \alpha, o)\} \\ E_2 & = & \{(\langle e,e\rangle, \sigma, o), (\langle e,o\rangle, \sigma, e), (\langle o,o\rangle, \sigma, o), (\langle o,e\rangle, \sigma, e)\} \end{array}$$

Then $L(\mathfrak{A}) = \{t \in \Sigma^{\circledast}[3,1] \mid \mathit{size}(t) \text{ is even}\}$.
To see this, consider for example $t = \langle\langle\rangle_3\alpha, \pi_2^3\rangle\sigma$. Then $\mathit{size}(t) = 2$, which is even. The following calculation shows that indeed $t \in L(\mathfrak{A})$.
First, $(\langle\rangle, \alpha, o) \in E_0$. Therefore, for every $Q_1, Q_2, Q_3 \subseteq Q$, we have

$$(\langle Q_1, Q_2, Q_3\rangle, \langle\rangle_3\alpha, o) \in E_{3,1}^{\mathfrak{A}}.$$

Therefore, in particular $(\langle\{e\},\{e\},\{e\}\rangle, \langle\rangle_3\alpha, o) \in E_{3,1}^{\mathfrak{A}}$. Next, by definition of $E_{3,1}^{\mathfrak{A}}$, we have $(\langle\{e\},\{e\},\{e\}\rangle, \pi_2^3, e) \in E_{3,1}^{\mathfrak{A}}$ because $e \in \{e\}$. This means that

$$(\langle\{e\},\{e\},\{e\}\rangle, \langle\langle\rangle_3\alpha, \pi_2^3\rangle, \langle o, e\rangle) \in E_{3,2}^{\mathfrak{A}}.$$

Since $(\langle o,e\rangle, \sigma, e) \in E_2$, it follows that

$$(\langle\{e\},\{e\},\{e\}\rangle, \langle\langle\rangle_3\alpha, \pi_2^3\rangle\sigma, e) \in E_{3,1}^{\mathfrak{A}}.$$

Since $e \in F$, this means that $t = \langle\langle\rangle_3\alpha, \pi_2^3\rangle\sigma \in L(\mathfrak{A})$.

5.6 Definition Let Σ be a ranked alphabet and $n, m \in \mathbb{N}$. A (n,m)-forest language $L \subseteq \Sigma^{\circledast}[n,m]$ is called *recognizable* if there is an (n,m)-fsfa $\mathfrak{A} = \langle Q, \Sigma, I, F, E\rangle$ with $L(\mathfrak{A}) = L$. The set of all recognizable (n,m)-forest languages over Σ is denoted by $\mathit{Rec}(\Sigma^{\circledast}, n, m)$.

5.7 Remark In the literature (e.g. [GS97], [Eng75]), the concept of finite-state tree automata is usually introduced by means of term rewriting systems. Such a tree automaton is "essentially the same" as a $(0,1)$-fsfa (as defined in Definition 5.1) because a $(0,1)$-fsfa has no initial states, the set of final state tuples is reduced to a set of final states, and the relations E_k contain exactly the same information as the rules of the term rewriting system of a tree automaton. If, for instance, $(\langle q_1, \ldots, q_k\rangle, \sigma, q) \in E_k$ (with $\sigma \in \Sigma_k$), then a bottom-up finite-state tree automaton would contain the rule $\sigma(q_1, \ldots, q_k) \to q$ and a top-down finite state tree automaton would contain the rule $q(\sigma(x_1, \ldots, x_k)) \to \sigma(q_1(x_1), \ldots, q_k(x_k))$. This means that for a given ranked alphabet Σ, the set $\mathit{Rec}(\Sigma^{\circledast}, 0, 1)$ is isomorphic to the set of recognizable tree languages over Σ. Sometimes (e.g. [GS97]) the discussion is casted in terms of tree languages over a ranked alphabet Σ and a finite set of variables X. In [GS97, p.9], the set recognizable tree languages over Σ and X is denoted by $Rec(\Sigma, X)$. If $n = |X|$ is the number of symbols in X, then $Rec(\Sigma, X)$ is isomorphic to $Rec(\Sigma^{\circledast}, n, 1)$.

5.8 An (n, m)-fsfa $\mathfrak{A} = \langle Q, \Sigma, I, F, E\rangle$ is called *deterministic* if $|I| = 1$ (i.e. there is only one input state tuple), and for every $k \geq 0$, every $\sigma \in \Sigma_k$ and every $q_1, \ldots, q_k, p_1, p_2 \in Q$, if $(\langle q_1, \ldots, q_k\rangle, \sigma, p_1) \in E_k$ and $(\langle q_1, \ldots, q_k\rangle, \sigma, p_2) \in E_k$, then $p_1 = p_2$ (i.e. the next state is uniquely determined).
An (n, m)-fsfa $\mathfrak{A} = \langle Q, \Sigma, I, F, E\rangle$ is called *totally defined* if for every $k \geq 0$, every $\sigma \in \Sigma_k$ and $q_1, \ldots, q_k \in Q$, there is a $p \in Q$ such that $(\langle q_1, \ldots, q_k\rangle, \sigma, p) \in E_k$.
For a deterministic and totally defined (n, m)-fsfa $\mathfrak{A} = \langle Q, \Sigma, I, F, E\rangle$, the relations $E_k \subseteq Q^k \times \Sigma_k \times Q$ are graphs of functions $\delta_k : Q^k \times \Sigma_k \to Q$. The following two definitions are based on this observation.

5.9 Definition Let $n, m \in \mathbb{N}$. A *deterministic, totally defined finite-state (n, m) forest automaton* (*(n, m)-dfsfa*) is a tuple $\mathfrak{A} = \langle Q, \Sigma, \vec{q}_I, F, \Delta\rangle$, where Q is a finite set of *states*, Σ is a ranked alphabet, $\vec{q}_I = \langle q_{I1}, \ldots, q_{In}\rangle \in Q^n$ is the *initial state tuple*, $F = F_1 \times \ldots \times F_m \subseteq Q^m$ is the set of *final state tuples*, and

$$\Delta = \{\delta_k \mid k \geq 0\}, \quad \text{where} \quad \delta_k : Q^k \times \Sigma_k \to Q \quad (\text{for every } k \geq 0)$$

is the set of *state transition functions*.

5.10 Remark Observe that for almost all $k \in \mathbb{N}$, the function $\delta_k : Q^k \times \Sigma_k \to Q$ is empty because Σ is a finite set (i.e. for almost all $k \in \mathbb{N}$, the set Σ_k is empty, cf. Remarks 3.4 and 5.2).

5.11 The behavior of an usual deterministic finite-state automaton on words over an alphabet A is determined by a function $\delta : Q \times A \to Q$, where Q is the set of states. This function can extended to a function $\hat{\delta} : Q \times A^* \to Q$, where $\delta(q_1, w) = q_2$ means that whenever the automaton is in state q_1, the input of word $w \in A^*$ causes a transition to state q_2 (cf. [Per90]). In the following paragraph, this idea is extended to deterministic finite-state forest automata. For every dfsfa $\mathfrak{A}$ and every $k, l \geq 0$, I will define a function $\delta^{\mathfrak{A}}_{k,l} : Q^k \times \Sigma^{\circledast}[k, l] \to Q^l$, where Q is the set of states of $\mathfrak{A}$. Informally speaking, $\delta^{\mathfrak{A}}_{k,l}(\langle q_1, \ldots, q_k\rangle, t) = \langle p_1, \ldots, p_l\rangle$ means that the forest $t \in \Sigma^{\circledast}[k, l]$ causes a transition from state tuple $\langle q_1, \ldots, q_k\rangle$ to state tuple $\langle p_1, \ldots, p_l\rangle$. Formally, this is made precise as follows.

5.12 For a given dfsfa $\mathfrak{A} = \langle Q, \Sigma, \vec{q}_I, F, \Delta\rangle$ with $\Delta = \{\delta_k \mid k \geq 0\}$, define for every $k, l \geq 0$ a function $\delta^{\mathfrak{A}}_{k,l} : Q^k \times \Sigma^{\circledast}[k, l] \to Q^l$ inductively as follows.

(i) For every $k \geq 0$ and $q_1, \ldots, q_k \in Q$,

$$\delta^{\mathfrak{A}}_{k,0}(\langle q_1, \ldots, q_k\rangle, \langle\rangle_k) = \langle\rangle,$$

where $\langle\rangle_k \in \Sigma^{\circledast}[k, 0]$ is the empty forest (cf. Definition 4.20) and $\langle\rangle \in Q^0$ is the empty tuple of states (cf. 2.1).

(ii) For every $k \geq 0$, every $q_1, \ldots, q_k \in Q$, and every $i = 1, \ldots, k$,

$$\delta^{\mathfrak{A}}_{k,1}(\langle q_1, \ldots, q_k\rangle, \pi^k_i) = q_i.$$

(iii) For every $k, k' \geq 0$, every $q_1, \ldots, q_k \in Q$, every $t \in \Sigma^{\circledast}[k, k']$ and every $\sigma \in \Sigma_{k'}$,

$$\delta^{\mathfrak{A}}_{k,1}(\langle q_1, \ldots, q_k\rangle, t\sigma) = \delta_{k'}(\delta^{\mathfrak{A}}_{k,k'}(\langle q_1, \ldots, q_k\rangle, t), \sigma).$$

(iv) For every $k \geq 0$ and $l \geq 2$, every $q_1, \ldots, q_k \in Q$, and every $t_1, \ldots, t_l \in \Sigma^{\circledast}[k, 1]$,

$$\delta^{\mathfrak{A}}_{k,l}(\langle q_1, \ldots, q_k \rangle, \langle t_1, \ldots, t_l \rangle) = \langle \delta^{\mathfrak{A}}_{k,1}(\langle q_1, \ldots, q_k \rangle, t_1), \ldots, \delta^{\mathfrak{A}}_{k,1}(\langle q_1, \ldots, q_k \rangle, t_l) \rangle.$$

5.13 Definition Let $\mathfrak{A} = \langle Q, \Sigma, \vec{q}_I, F, \Delta \rangle$ be an (n,m)-dfsfa. Then

$$L(\mathfrak{A}) = \{t \in \Sigma^{\circledast}[n, m] \mid \delta^{\mathfrak{A}}_{n,m}(\vec{q}_I, t) \in F\}$$

is the language *accepted* (or *recognized*) by $\mathfrak{A}$.

5.14 Example Consider $\Sigma = \Sigma_0 \cup \Sigma_2$ with $\Sigma_0 = \{\alpha\}$ and $\Sigma_2 = \{\sigma\}$. Define a $(3,1)$-dfsfa $\mathfrak{A} = \langle Q, \Sigma, \vec{q}_I, F, \Delta \rangle$ as follows.

$$\begin{array}{rcl}
Q & = & \{e, o\} \\
\vec{q}_I & = & \langle e, e, e \rangle \\
F & = & \{e\} \\
\delta_0(\langle\rangle, \alpha) & = & o \\
\delta_2(\langle e, e \rangle, \sigma) & = & o \\
\delta_2(\langle e, o \rangle, \sigma) & = & e \\
\delta_2(\langle o, e \rangle, \sigma) & = & e \\
\delta_2(\langle o, o \rangle, \sigma) & = & o
\end{array}$$

Then, $L(\mathfrak{A}) = \{t \in \Sigma^{\circledast}[3, 1] \mid \mathit{size}(t) \text{ is even}\}$. Let me use as an example the same forest $t = \langle \langle\rangle_3 \alpha, \pi^3_2 \rangle \sigma$ as in Example 5.5.

$$\begin{array}{rcl}
\delta^{\mathfrak{A}}_{3,1}(\vec{q}_I, t) & = & \delta^{\mathfrak{A}}_{3,1}(\langle e, e, e \rangle, \langle \langle\rangle_3 \alpha, \pi^3_2 \rangle \sigma) \\
& = & \delta_2(\delta^{\mathfrak{A}}_{3,2}(\langle e, e, e \rangle, \langle \langle\rangle_3 \alpha, \pi^3_2 \rangle), \sigma) \\
& = & \delta_2(\langle \delta^{\mathfrak{A}}_{3,1}(\langle e, e, e \rangle, \langle\rangle_3 \alpha), \delta^{\mathfrak{A}}_{3,1}(\langle e, e, e \rangle, \pi^3_2) \rangle, \sigma) \\
& = & \delta_2(\langle \delta_0(\delta^{\mathfrak{A}}_{3,0}(\langle e, e, e \rangle, \langle\rangle_3), \alpha), e \rangle, \sigma) \\
& = & \delta_2(\langle \delta_0(\langle\rangle, \alpha), e \rangle, \sigma) \\
& = & \delta_2(\langle o, e \rangle, \sigma) \\
& = & e
\end{array}$$

Since $e \in F$, it is indeed the case that $t \in L(\mathfrak{A})$.

5.15 Remark Similarly as in the nondeterministic case, a (0,1)-dfsfa can be seen as a deterministic fsta, which is usually (e.g. [TW68] or [GS97]) defined as a finite Σ-algebra with underlying set Q. However, it is possible, to consider any (n,m)-dfsfa $\mathfrak{A} = \langle Q, \Sigma, \vec{q}_I, F, \Delta \rangle$ as a finite Σ-algebra $\mathcal{Q}_{\mathfrak{A}}$ with underlying set Q. This means, that for every $k \geq 0$ and $\sigma \in \Sigma_k$, there is a function $\sigma^Q : Q^k \to Q$. This function is (in my definition) carried out by the function δ_k. If $\delta_k(\langle q_1, \ldots, q_k \rangle, \sigma) = q$, then $\sigma^Q(q_1, \ldots, q_k) = q$. Furthermore, every function $g : X_n \to Q$ from a set X_n of n variables into the set Q can be extended to a unique homomorphism $\hat{g} : T_\Sigma(X_n) \to Q$. In my definition, the initial state tuple $\vec{q}_I$ contains the information of g, and the function $\delta^{\mathfrak{A}}_{n,1}$ carries out the homomorphism $\hat{g}$ if it gets $\vec{q}_I$ as first input value. Taking the isomorphism $\varphi : \Sigma^{\circledast}[n, 1] \to T_\Sigma(X_n)$ from Proposition 4.27, this spells out as

$$\delta^{\mathfrak{A}}_{n,1}(\vec{q}_I, t) = \hat{g}(\varphi(t))$$

for every $t \in \Sigma^{\circledast}[n,1]$. I will not go into further details here, because they are not needed for my work. However, for some constructions it might be helpful to think of an automaton as a finite Σ-algebra.

5.16 Let me define $d\text{-}Rec(\Sigma^{\circledast}, n, m)$ to be the set of languages recognizable by an (n,m)-dfsfa. Then it follows immediately from the definitions that $d\text{-}Rec(\Sigma^{\circledast}, n, m) \subseteq Rec(\Sigma^{\circledast}, n, m)$. The following proposition says that $Rec(\Sigma^{\circledast}, n, m) \subseteq d\text{-}Rec(\Sigma^{\circledast}, n, m)$. Therefore, both sets are equal. The proof uses the well-known power set construction from automata theory.

5.17 Proposition *Let $n, m \in \mathbb{N}$ and $\mathfrak{A}$ be an (n,m)-fsfa. Then there exists an (n,m)-dfsfa $\mathfrak{A}'$ such that*

$$L(\mathfrak{A}') = L(\mathfrak{A}).$$

Proof: Let $\mathfrak{A} = \langle Q, \Sigma, I, F, E\rangle$ with $I = I_1 \times \ldots \times I_n$ and $E = \{E_k \mid k \in \mathbb{N}\}$. Define $\mathfrak{A}' = \langle Q', \Sigma, \vec{q_I}', F', \Delta\rangle$ with $\Delta = \{\delta_k \mid k \in \mathbb{N}\}$ as follows.

$$\begin{array}{rcl} Q' &=& \mathfrak{P}(Q) \\ \vec{q_I}' &=& \langle I_1, \ldots, I_n\rangle \\ F' &=& \{\langle Q_1, \ldots, Q_m\rangle \in (\mathfrak{P}(Q))^m \mid \exists \langle q_1 \ldots, q_m\rangle \in F \;.\; q_1 \in Q_1 \wedge \ldots \wedge q_m \in Q_m\} \\ \delta_k(\langle Q_1, \ldots, Q_k\rangle, \sigma) &=& \{p \in Q \mid \exists q_1 \in Q_1 \ldots \exists q_k \in Q_k \;.\; (\langle q_1, \ldots, q_k\rangle, \sigma, p) \in E_k\} \\ && \text{(for every } k \geq 0,\ Q_1, \ldots, Q_k \subseteq Q,\ \sigma \in \Sigma_k) \end{array}$$

Now it remains to show that indeed $L(\mathfrak{A}) = L(\mathfrak{A}')$. In order to do so, let me prove the following stronger statement, which exhibits an essential property of the (n,m)-dfsfa $\mathfrak{A}'$.

Claim 1: For every $t \in \Sigma^{\circledast}[n,1]$,

$$\delta^{\mathfrak{A}'}_{n,1}(\langle I_1, \ldots, I_n\rangle, t) = \{p \in Q \mid (\langle I_1, \ldots, I_n\rangle, t, p) \in E^{\mathfrak{A}}_{n,1}\}.$$

Proof of Claim 1: The proof will be done by induction on the structure of t.

- $\underline{t = \pi^n_i}$ (for some $i = 1, \ldots, n$). Then

$$\begin{array}{rcl} \delta^{\mathfrak{A}'}_{n,1}(\langle I_1, \ldots, I_n\rangle, \pi^n_i) &=& I_i \quad \text{(by definition of } \delta^{\mathfrak{A}}_{n,1} \text{ in 5.12)} \\ &=& \{p \in Q \mid (\langle I_1, \ldots, I_n\rangle, \pi^n_i, p) \in E^{\mathfrak{A}}_{n,1}\} \quad \text{(by definition of } E^{\mathfrak{A}}_{n,1} \text{ in 5.3)} \end{array}$$

- $\underline{t = t'\sigma}$ (for some $k \geq 0$, $\sigma \in \Sigma_k$, $t' = \langle t'_1, \ldots, t'_k\rangle \in \Sigma^{\circledast}[n,k]$).
  - For the first inclusion let $p \in \delta^{\mathfrak{A}'}_{n,1}(\langle I_1, \ldots, I_n\rangle, t'\sigma) = \delta_k(\delta^{\mathfrak{A}'}_{n,k}(\langle I_1, \ldots, I_n\rangle, t'), \sigma)$. Furthermore, let $\langle P_1, \ldots, P_k\rangle = \delta^{\mathfrak{A}'}_{n,k}(\langle I_1, \ldots, I_n\rangle, \langle t'_1, \ldots, t'_k\rangle)$, where $P_i \subseteq Q$ for $i = 1, \ldots, k$. Then there is a tuple $\langle p_1, \ldots, p_k\rangle$ with $p_i \in P_i$ for every $i = 1, \ldots, k$, such that $(\langle p_1, \ldots, p_k\rangle, \sigma, p) \in E_k$ (by definition of δ_k above). By induction hypothesis, for every $i = 1, \ldots, k$, we have $(\langle I_1, \ldots, I_n\rangle, t'_i, p_i) \in E^{\mathfrak{A}}_{n,1}$. Hence, $(\langle I_1, \ldots, I_n\rangle, t', \langle p_1, \ldots, p_k\rangle) \in E^{\mathfrak{A}}_{n,k}$. It follows (see 5.3) that $(\langle I_1, \ldots, I_n\rangle, t'\sigma, p) \in E^{\mathfrak{A}}_{n,1}$.
  - For the other direction, assume that $(\langle I_1, \ldots, I_n\rangle, t'\sigma, p) \in E^{\mathfrak{A}}_{n,1}$. This means, that there is a tuple $\langle p_1, \ldots, p_k\rangle$ with $(\langle p_1, \ldots, p_k\rangle, \sigma, p) \in E_k$ and $(\langle I_1, \ldots, I_n\rangle, t', \langle p_1, \ldots, p_k\rangle) \in E^{\mathfrak{A}}_{n,k}$, i.e. , for every $i = 1, \ldots, k$, we have $(\langle I_1, \ldots, I_n\rangle, t'_i, p_i) \in E^{\mathfrak{A}}_{n,1}$.
  Now let $\langle P_1, \ldots, P_k\rangle = \delta^{\mathfrak{A}'}_{n,k}(\langle I_1, \ldots, I_n\rangle, t')$. Then (by induction hypothesis) $p_i \in P_i$ for every $i = 1, \ldots, k$, and therefore $p \in \delta_k(\langle P_1, \ldots, P_k\rangle, \sigma)$ by definition of δ_k. Hence, $p \in \delta_k(\delta^{\mathfrak{A}'}_{n,k}(\langle I_1, \ldots, I_k\rangle, t'), \sigma) = \delta^{\mathfrak{A}'}_{n,1}(\langle I_1, \ldots, I_k\rangle, t'\sigma)$.

This completes the proof of the claim.

Now let me use this statement to show that the automata $\mathfrak{A}$ and $\mathfrak{A}'$ recognize the same language. First, let $t = \langle t_1, \dots, t_m \rangle \in L(\mathfrak{A}')$. Then $\delta^{\mathfrak{A}'}_{n,m}(\langle I_1, \dots, I_n \rangle, t) = \langle P_1, \dots, P_m \rangle \in F'$. By definition of F', there is a tuple $\langle p_1, \dots, p_m \rangle \in F$ with $p_i \in P_i$ for every $i = 1, \dots, m$. By the claim, we have that $(\langle I_1, \dots, I_n \rangle, t_i, p_i) \in E^{\mathfrak{A}}_{n,1}$ for every $i = 1, \dots, m$. Hence, $(\langle I_1, \dots, I_n \rangle, t, \langle p_1, \dots, p_m \rangle) \in E^{\mathfrak{A}}_{n,m}$. Therefore, $t \in L(\mathfrak{A})$. Conversely, let $t = \langle t_1, \dots, t_m \rangle \in L(\mathfrak{A})$. Then, there is a tuple $\langle p_1, \dots, p_m \rangle \in F$ such that $(\langle I_1, \dots, I_n \rangle, \langle t_1, \dots, t_m \rangle, \langle p_1, \dots, p_m \rangle) \in E^{\mathfrak{A}}_{n,m}$. By the claim, for every $i = 1, \dots, m$, we have $p_i \in \delta^{\mathfrak{A}'}_{n,1}(\langle I_1, \dots, I_n \rangle, t_i)$. Since $\langle p_1, \dots, p_m \rangle \in F$, it follows (by definition of F') that $\delta^{\mathfrak{A}'}_{n,m}(\langle I_1, \dots, I_n \rangle, t) \in F'$. Therefore $t \in L(\mathfrak{A}')$. Hence $L(\mathfrak{A}) = L(\mathfrak{A}')$. □

5.18 Corollary *If Σ is a ranked alphabet and n, m are natural numbers, then $\emptyset \in Rec(\Sigma^{\circledast}, n, m)$ and $\Sigma^{\circledast}[n, m] \in Rec(\Sigma^{\circledast}, n, m)$.*

Proof: For the empty set choose an (n, m)-dfsfa $\mathfrak{A} = \langle Q, \Sigma, \vec{q}_I, F, \Delta \rangle$ (where $F = F_1 \times \ldots \times F_m$) with $F_i = \emptyset$ (for every $i = 1, \dots, m$). For $\Sigma^{\circledast}[n, m]$, let $F_i = Q$ (for every $i = 1, \dots, m$).
Some attention deserves the case $m = 0$. By definition, $\Sigma^{\circledast}[n, 0] = \{\langle\rangle_n\}$. If $\mathfrak{A} = \langle Q, \Sigma, \vec{q}_I, F, \Delta \rangle$ is an $(n, 0)$-dfsfa, then $F \subseteq Q^0$. So, either $F = \emptyset$ or $F = Q^0 = \{\langle\rangle\}$. In the first case $L(\mathfrak{A}) = \emptyset$, and in the latter $L(\mathfrak{A}) = \{\langle\rangle_n\} = \Sigma^{\circledast}[n, 0]$. □

5.19 Observe that for a given ranked alphabet Σ and natural numbers n_1, n_2, m_1, m_2, if $n_1 \neq n_2$ or $m_1 \neq m_2$, then

$$Rec(\Sigma^{\circledast}, n_1, m_1) \cap Rec(\Sigma^{\circledast}, n_2, m_2) = \{\emptyset\}.$$

Therefore (see 2.4 and 4.1), by removing the empty set, we can obtain an $\mathbb{N} \times \mathbb{N}$-indexed set $Rec(\Sigma^{\circledast})$. Formally,

$$Rec(\Sigma^{\circledast}) = \bigcup_{n,m \in \mathbb{N}} Rec(\Sigma^{\circledast}, n, m) \setminus \{\emptyset\}$$

is an $\mathbb{N} \times \mathbb{N}$-indexed set where $Rec(\Sigma^{\circledast})[n, m] = Rec(\Sigma^{\circledast}, n, m) \setminus \{\emptyset\}$ for every $n, m \in \mathbb{N}$.

5.20 Proposition *Let Σ be a ranked alphabet and $n, m \in \mathbb{N}$. Then, a language $L \subseteq \Sigma^{\circledast}[n, m]$ is recognizable iff there exist a finite polyoid $\mathcal{P} = \langle P, ;_{\mathcal{P}}, \langle \ldots \rangle_{\mathcal{P}}, \Pi_{\mathcal{P}} \rangle$, a polyoid homomorphism $h : \Sigma^{\circledast} \to P$, and subsets $G_1, \dots, G_m \subseteq P[n, 1]$ such that $L = h^{-1}(G_1) \times \ldots \times h^{-1}(G_m)$.*

Proof: Let me first show that the condition is necessary. Consider an (n, m)-dfsfa $\mathfrak{A} = \langle Q, \Sigma, \vec{q}_I, F, \Delta \rangle$ with $F = F_1 \times \ldots \times F_m$ such that $L(\mathfrak{A}) = L$. Let $\mathcal{P}$ be the polyoid $\mathcal{P}_{Fun}Q$ from Example 4.10. Then, $\mathcal{P}$ is finite because Q is finite. Now define a function $h : \Sigma^{\circledast} \to \mathrm{Fun}(Q)$ as follows. For $k, l \in \mathbb{N}$ and every forest $t \in \Sigma^{\circledast}[k, l]$ define a function $\bar{t} : Q^k \to Q^l$ which takes an k-tuple $\langle q_1, \dots, q_k \rangle$ of states to the l-tuple $\delta^{\mathfrak{A}}_{k,l}(\langle q_1, \dots, q_k \rangle, t)$ and let $h(t) = \bar{t}$. Then, h is a polyoid homomorphism because, by the definition of $\delta^{\mathfrak{A}}_{k,l}$ in 5.12, h fulfills Conditions (i), (iii), and (iv) of Definition 4.15, and by an easy induction it can be shown that also (ii) does hold. Next, define for every $i = 1, \dots, m$, the set $G_i = \{g \in \mathrm{Fun}(Q)[n, 1] \mid g(\vec{q}_I) \in F_i\}$. Then, for every $t = \langle t_1, \dots, t_m \rangle \in \Sigma^{\circledast}[n, m]$,

$$\begin{aligned}
t \in L \quad &\iff \quad \delta^{\mathfrak{A}}_{n,m}(\vec{q}_I, t) \in F \\
&\iff \quad \forall i \in \{1, \dots, m\} \,.\, \delta^{\mathfrak{A}}_{n,1}(\vec{q}_I, t_i) \in F_i \\
&\iff \quad \forall i \in \{1, \dots, m\} \,.\, h(t_i)(\vec{q}_I) \in F_i \\
&\iff \quad \forall i \in \{1, \dots, m\} \,.\, h(t_i) \in G_i \\
&\iff \quad \forall i \in \{1, \dots, m\} \,.\, t_i \in h^{-1}(G_i) \\
&\iff \quad t \in h^{-1}(G_1) \times \ldots \times h^{-1}(G_m).
\end{aligned}$$

Thus, $L = h^{-1}(G_1) \times \ldots \times h^{-1}(G_m)$.

Conversely, let $\mathcal{P} = \langle P, ;_{\mathcal{P}}, \langle\ldots\rangle_{\mathcal{P}}, \Pi_{\mathcal{P}}\rangle$ be a finite polyoid, $h : \Sigma^{\circledast} \to P$ a homomorphism, and $G_1, \ldots, G_m \subseteq P[n,1]$ be given such that $L = h^{-1}(G_1) \times \ldots \times h^{-1}(G_m)$. Define an (n,m)-dfsfa $\mathfrak{A} = \langle Q, \Sigma, \vec{q}_I, F, \Delta\rangle$ as follows. Let

$$\begin{aligned} Q &= P[n,1], \\ \vec{q}_I &= \langle {\pi_1^n}_{\mathcal{P}}, \ldots, {\pi_n^n}_{\mathcal{P}}\rangle, \\ F &= G_1 \times \ldots \times G_m \quad \text{(i.e. } F_i = G_i \text{ for every } i = 1, \ldots, m), \end{aligned}$$

and define for every $k \geq 0$, the function δ_k by

$$\delta_k(\langle q_1, \ldots, q_k\rangle, \sigma) = \langle\!\langle q_1, \ldots, q_k\rangle\!\rangle_{\mathcal{P}} ;_{\mathcal{P}} h(\varepsilon_k \sigma)$$

for every $q_1, \ldots, q_k \in P[n,1]$ and $\sigma \in \Sigma_k$. It remains to show that $L(\mathfrak{A}) = L$. Since h is a homomorphism, we have $\delta^{\mathfrak{A}}_{k,1}(\langle q_1, \ldots, q_k\rangle, t) = \langle\!\langle q_1, \ldots, q_k\rangle\!\rangle_{\mathcal{P}} ;_{\mathcal{P}} h(t)$ for every $t \in \Sigma^{\circledast}[k,1]$. Hence, for every $t = \langle t_1, \ldots, t_m\rangle \in \Sigma^{\circledast}[n,m]$,

$$\begin{aligned} t \in L(\mathfrak{A}) &\iff \delta^{\mathfrak{A}}_{n,m}(\vec{q}_I, t) \in F \\ &\iff \forall i \in \{1, \ldots, m\} \,.\, \delta^{\mathfrak{A}}_{n,1}(\vec{q}_I, t_i) \in G_i \\ &\iff \forall i \in \{1, \ldots, m\} \,.\, \vec{q}_I ;_{\mathcal{P}} h(t_i) \in G_i \\ &\iff \forall i \in \{1, \ldots, m\} \,.\, h(t_i) \in G_i \\ &\iff \forall i \in \{1, \ldots, m\} \,.\, t_i \in h^{-1}(G_i) \\ &\iff t \in h^{-1}(G_1) \times \ldots \times h^{-1}(G_m). \end{aligned}$$

Therefore, $L(\mathfrak{A}) = h^{-1}(G_1) \times \ldots \times h^{-1}(G_m) = L$. □

5.21 Remark Proposition 5.20 generalizes an important characterization of recognizable string languages. In [Ber79, p.52], this property is used to define the notion of recognizability in every monoid. In a similar way, it is possible to define for every polyoid $\mathcal{P}$ the class $Rec(\mathcal{P})$ of recognizable sets in $\mathcal{P}$. However, the investigation of the properties of recognizable sets in any polyoid would take us too far beyond the scope of this thesis. For that reason, I will restrict myself to the case of the free polyoid.

6 Closure Properties of Recognizable Forest Languages

In this section, I will show some basic properties of the recognizable forest languages, which have been defined in the previous section. As already mentioned in the previous section, the set $Rec(\Sigma^{\circledast}, n, 1) \subseteq \mathfrak{P}(\Sigma^{\circledast}[n,1])$ of recognizable $(n,1)$-forest languages over Σ is isomorphic to the set $Rec(\Sigma, X_n) \subseteq \mathfrak{P}(T_\Sigma(X_n))$, of recognizable tree languages (cf. [GS97]). Hence, $Rec(\Sigma^{\circledast}, n, 1)$ is closed under the boolean operators intersection, union, and complement. However, because of the requirement that the set F of final state tuples of an (n,m)-fsfa $\mathfrak{A} = \langle Q, \Sigma, I, F, E\rangle$ has to be of the form $F = F_1 \times \ldots \times F_m$, the set $Rec(\Sigma^{\circledast}, n, m)$ has in general not the same closure properties as $Rec(\Sigma^{\circledast}, n, 1)$.

6.1 Proposition *Let Σ be a ranked alphabet and n, m be natural numbers. Furthermore, let $L_1, L_2 \in Rec(\Sigma^{\circledast}, n, m)$ be recognizable sets. Then,*

(i) $L_1 \cap L_2 \in Rec(\Sigma^{\circledast}, n, m)$, and

(ii) if $m = 1$, then $L_1^C = \Sigma^{\circledast}[n,m] \setminus L_1$ as well as $L_1 \cup L_2$ and $L_1 \setminus L_2$ are recognizable.

Proof: (i) Let $\mathfrak{A}_1 = \langle Q_1, \Sigma, \vec{q}_{I1}, F_1, \Delta_1 \rangle$ with $\Delta_1 = \{\delta_{1k} \mid k \in \mathbb{N}\}$ and $\mathfrak{A}_2 = \langle Q_2, \Sigma, \vec{q}_{I2}, F_2, \Delta_2 \rangle$ with $\Delta_2 = \{\delta_{2k} \mid k \in \mathbb{N}\}$ be (n,m)-dfsfa's with $L(\mathfrak{A}_1) = L_1$ and $L(\mathfrak{A}_2) = L_2$. For disambiguating the indices, let $\vec{q}_{I1} = \langle q_{I11}, \ldots, q_{I1n} \rangle$, $\vec{q}_{I2} = \langle q_{I21}, \ldots, q_{I2n} \rangle$, $F_1 = F_{11} \times \ldots \times F_{1m}$, and $F_2 = F_{21} \times \ldots \times F_{2m}$. Now construct the product automaton $\mathfrak{A}_\times = \langle Q_\times, \Sigma, \vec{q}_{I\times}, F_\times, \delta_\times \rangle$ as follows.

$$
\begin{array}{rcl}
Q_\times & = & Q_1 \times Q_2 \\
\vec{q}_{I\times} & = & \langle \langle q_{I11}, q_{I21} \rangle, \ldots, \langle q_{I1n}, q_{I2n} \rangle \rangle \\
F_\times & = & (F_{11} \times F_{21}) \times \ldots \times (F_{1m} \times F_{2m}) \\
\delta_{\times k}(\langle \langle q_{11}, q_{21} \rangle, \ldots, \langle q_{1k}, q_{2k} \rangle \rangle, \sigma) & = & \langle \delta_{1k}(\langle q_{11}, \ldots, q_{1k} \rangle, \sigma), \delta_{2k}(\langle q_{21}, \ldots, q_{2k} \rangle, \sigma) \rangle \text{ for every } k \geq 0.
\end{array}
$$

Claim 1: For every $k \geq 0$, $t \in \Sigma^{\circledast}[k,1]$, $q_{11}, \ldots, q_{1k} \in Q_1$, and $q_{21}, \ldots, q_{2k} \in Q_2$,

$$
\delta_{k,1}^{\mathfrak{A}_\times}(\langle \langle q_{11}, q_{21} \rangle, \ldots, \langle q_{1k}, q_{2k} \rangle \rangle, t) = \langle \delta_{k,1}^{\mathfrak{A}_1}(\langle q_{11}, \ldots, q_{1k} \rangle, t), \delta_{k,1}^{\mathfrak{A}_2}(\langle q_{21}, \ldots, q_{2k} \rangle, t) \rangle.
$$

Proof of Claim 1: By induction on the structure of t.

- $\underline{t = \pi_i^k}$ (for some $i = 1, \ldots, k$). Then

$$
\begin{array}{rcl}
\delta_{k,1}^{\mathfrak{A}_\times}(\langle \langle q_{11}, q_{21} \rangle, \ldots, \langle q_{1k}, q_{2k} \rangle \rangle, \pi_i^k) & = & \langle q_{1i}, q_{2i} \rangle \\
& = & \langle \delta_{k,1}^{\mathfrak{A}_1}(\langle q_{11}, \ldots, q_{1k} \rangle, \pi_i^k), \delta_{k,1}^{\mathfrak{A}_2}(\langle q_{21}, \ldots, q_{2k} \rangle, \pi_i^k) \rangle
\end{array}
$$

- $\underline{t = \langle t_1, \ldots, t_k \rangle \sigma}$ (for some $k' \geq 0$, $\sigma \in \Sigma_{k'}$, $t_1, \ldots, t_{k'} \in \Sigma^{\circledast}[k,1]$). Then

$$
\begin{array}{rl}
& \delta_{k,1}^{\mathfrak{A}_\times}(\langle \langle q_{11}, q_{21} \rangle, \ldots, \langle q_{1k}, q_{2k} \rangle \rangle, \langle t_1, \ldots, t_{k'} \rangle \sigma) \\
= & \delta_{\times k}(\langle \delta_{k,1}^{\mathfrak{A}_\times}(\langle \langle q_{11}, q_{21} \rangle, \ldots, \langle q_{1k}, q_{2k} \rangle \rangle, t_1), \ldots, \delta_{k,1}^{\mathfrak{A}_\times}(\langle \langle q_{11}, q_{21} \rangle, \ldots, \langle q_{1k}, q_{2k} \rangle \rangle, t_{k'}) \rangle, \sigma) \\
& \text{(by 5.12)} \\
= & \delta_{\times k}(\langle \quad \langle \delta_{k,1}^{\mathfrak{A}_1}(\langle q_{11}, \ldots, q_{1k} \rangle, t_1), \delta_{k,1}^{\mathfrak{A}_2}(\langle q_{21}, \ldots, q_{2k} \rangle, t_1) \rangle, \ldots, \\
& \qquad \langle \delta_{k,1}^{\mathfrak{A}_1}(\langle q_{11}, \ldots, q_{1k} \rangle, t_{k'}), \delta_{k,1}^{\mathfrak{A}_2}(\langle q_{21}, \ldots, q_{2k} \rangle, t_{k'}) \rangle \quad \rangle, \sigma) \\
& \text{(by induction hypothesis)} \\
= & \langle \quad \delta_{1k}(\langle \delta_{k,1}^{\mathfrak{A}_1}(\langle q_{11}, \ldots, q_{1k} \rangle, t_1), \ldots, \delta_{k,1}^{\mathfrak{A}_1}(\langle q_{11}, \ldots, q_{1k} \rangle, t_{k'}) \rangle, \sigma), \\
& \qquad \delta_{2k}(\langle \delta_{k,1}^{\mathfrak{A}_2}(\langle q_{11}, \ldots, q_{1k} \rangle, t_1), \ldots, \delta_{k,1}^{\mathfrak{A}_2}(\langle q_{11}, \ldots, q_{1k} \rangle, t_{k'}) \rangle, \sigma) \quad \rangle \\
& \text{(by definition of } \delta_{\times k}\text{)} \\
= & \langle \delta_{k,1}^{\mathfrak{A}_1}(\langle q_{11}, \ldots, q_{1k} \rangle, \langle t_1, \ldots, t_k \rangle \sigma), \delta_{k,1}^{\mathfrak{A}_2}(\langle q_{21}, \ldots, q_{2k} \rangle, \langle t_1, \ldots, t_k \rangle \sigma) \rangle \\
& \text{(by 5.12)}
\end{array}
$$

This completes the proof of Claim 1.
Remark: Together with Remark 5.15, Claim 1 simply restates the well-known fact that the product of two finite Σ-algebras is again a finite Σ-algebra.

Using Claim 1, it follows that

$$
\begin{array}{rcl}
t \in L(\mathfrak{A}_\times) & \iff & \delta_{n,m}^{\mathfrak{A}_\times}(\vec{q}_{I\times}, t) \in F_\times \\
& \iff & \delta_{n,m}^{\mathfrak{A}_1}(\vec{q}_{I1}, t) \in F_1 \wedge \delta_{n,m}^{\mathfrak{A}_2}(\vec{q}_{I2}, t) \in F_2 \\
& \iff & t \in L(\mathfrak{A}_1) \wedge t \in L(\mathfrak{A}_2) \\
& \iff & t \in L_1 \cap L_2
\end{array}
$$

Hence $L(\mathfrak{A}_\times) = L_1 \cap L_2$.

(ii) Let $\mathfrak{A} = \langle Q, \Sigma, \vec{q}_I, F, \Delta\rangle$ be an $(n,1)$-dfsfa such that $L(\mathfrak{A}) = L_1$. Let $\mathfrak{A}^C = \langle Q, \Sigma, \vec{q}_I, F^C, \Delta\rangle$ where $F^C = Q \setminus F$. Then $L(\mathfrak{A}^C) = \Sigma^{\circledast}[n,1] \setminus L(\mathfrak{A}) = L(\mathfrak{A})^C = L_1^C$.
The closure under $\cup$ and $\setminus$ follow from this, since $L_1 \cup L_2 = (L_1^C \cap L_2^C)^C$ and $L_1 \setminus L_2 = L_1 \cap L_2^C$. $\square$

6.2 Lemma *If Σ is a ranked alphabet and n a natural numbers, then for every $t \in \Sigma^{\circledast}[n,1]$,*

$$\{t\} \in Rec(\Sigma^{\circledast}, n, 1).$$

In other words, $Rec(\Sigma^{\circledast}, n, 1)$ contains all singletons.

Proof: This fact follows immediately from the according property of the class $Rec(\Sigma, X_n)$ of recognizable tree languages (see e.g. [GS97]). For that reason, I will not give a detailed proof here. But it should be obvious that for a given $t \in \Sigma^{\circledast}[n,1]$ an $(n,1)$-fsfa $\mathfrak{A}$ can be constructed such that $L(\mathfrak{A}) = \{t\}$. $\square$

6.3 Remark Later (from Proposition 6.12) it will follow that for every $n, m \geq 0$, the class $Rec(\Sigma^{\circledast}, n, m)$ contains all singletons.

6.4 Corollary *For every ranked alphabet Σ and natural number n, the set $Rec(\Sigma^{\circledast}, n, 1)$ forms a boolean algebra, namely a subalgebra of $\mathfrak{P}(\Sigma^{\circledast}[n,1])$. Furthermore, it contains all finite subsets of $\Sigma^{\circledast}[n,1]$.*

Proof: It only remains to show that $Rec(\Sigma^{\circledast}, n, 1)$ contains all finite subsets of $\Sigma^{\circledast}[n,1]$. But this follows from the fact that the singletons are in $Rec(\Sigma^{\circledast}, n, 1)$ (Lemma 6.2), and that it is closed under union (Proposition 6.1). $\square$

6.5 It is important to notice, that the set $Rec(\Sigma^{\circledast}, n, m)$ is in general not closed under union. The following example will make clear, why this is the case. However, the recognizable languages are closed under cartesian product (see Proposition 6.12). With this as a base, I will define another operation $\uplus$ (in Definition 6.16), which will take the place of the union.

6.6 Example Consider again $\Sigma = \Sigma_0 \cup \Sigma_2$ with $\Sigma_0 = \{\alpha\}$ and $\Sigma_2 = \{\sigma\}$ from Example 5.5. Construct the following two $(1,2)$-fsfa's $\mathfrak{A}_a = \langle Q, \Sigma, I, F_a, E\rangle$ with

$$\begin{aligned}
Q &= \{e, o\} \\
I &= \{\langle e\rangle\} \\
F_a &= \{e\} \times \{o\} \;=\; \{\langle e, o\rangle\} \\
E_0 &= \{(\langle\rangle, \alpha, o)\} \\
E_2 &= \{(\langle e, e\rangle, \sigma, o), (\langle e, o\rangle, \sigma, e), (\langle o, o\rangle, \sigma, o), (\langle o, e\rangle, \sigma, e)\}
\end{aligned}$$

and $\mathfrak{A}_b = \langle Q, \Sigma, I, F_b, E\rangle$ with

$$\begin{aligned}
Q &= \{e, o\} \\
I &= \{\langle e\rangle\} \\
F_b &= \{o\} \times \{e\} \;=\; \{\langle o, e\rangle\} \\
E_0 &= \{(\langle\rangle, \alpha, o)\} \\
E_2 &= \{(\langle e, e\rangle, \sigma, o), (\langle e, o\rangle, \sigma, e), (\langle o, o\rangle, \sigma, o), (\langle o, e\rangle, \sigma, e)\}.
\end{aligned}$$

Then

$$\begin{aligned}
L(\mathfrak{A}_a) &= \{\langle t_1, t_2\rangle \in \Sigma^{\circledast}[3,2] \mid \mathit{size}(t_1) \text{ is even and } \mathit{size}(t_2) \text{ is odd}\} \quad \text{and} \\
L(\mathfrak{A}_b) &= \{\langle t_1, t_2\rangle \in \Sigma^{\circledast}[3,2] \mid \mathit{size}(t_1) \text{ is odd and } \mathit{size}(t_2) \text{ is even}\}.
\end{aligned}$$

This means that

$$L(\mathfrak{A}_a) \cup L(\mathfrak{A}_b) = \{\langle t_1, t_2\rangle \in \Sigma^{\circledast}[3,2] \mid \mathit{size}(t_1) + \mathit{size}(t_2) \text{ is odd}\},$$

which is not recognizable. After Proposition 6.12 it will become clear, why this is the case (see Example 6.14).
I will return to this example in Section 8, where I will consider the semi-recognizable languages that are closed under union.

6.7 Definition Let Σ be a ranked alphabet, let $n, m, k \in \mathbb{N}$, and let $L_1 \subseteq \Sigma^{\circledast}[n,m]$ and $L_2 \subseteq \Sigma^{\circledast}[m,k]$. Then

$$L_1; L_2 = \{t_1; t_2 \mid t_1 \in L_1, t_2 \in L_2\}$$

is the *naive concatenation* of L_1 and L_2.

6.8 In the case that L_2 is a singleton, say $L_2 = \{\nu\}$, I will write $L_1; \nu$ (instead of $L_1; \{\nu\}$) for notational convenience. Similarly, I will write $\nu; L_2$ instead of $\{\nu\}; L_2$.

6.9 Definition Let Σ be a ranked alphabet, and n, m_1, m_2 be natural numbers. Further, let $L_1 \subseteq \Sigma^{\circledast}[n, m_1]$ and $L_2 \subseteq \Sigma^{\circledast}[n, m_2]$. Then

$$L_1 \times L_2 = \{\langle t_1, \ldots, t_{m_1+m_2}\rangle \in \Sigma^{\circledast}[n, m_1 + m_2] \mid \langle t_1, \ldots, t_{m_1}\rangle \in L_1 \wedge \langle t_{m_1+1}, \ldots, t_{m_1+m_2}\rangle \in L_2\}$$

is the *cartesian product* of L_1 and L_2.

6.10 Remark The operation defined in 6.9 is associative. Furthermore, if Σ is a ranked alphabet, n, m are natural numbers and $L_1, \ldots, L_m \subseteq \Sigma^{\circledast}[n,1]$ are $(n,1)$-forest languages, then

$$L_1 \times \ldots \times L_m = \{\langle t_1, \ldots, t_m\rangle \in \Sigma^{\circledast}[n,m] \mid \forall i \in \{1, \ldots, m\} \,.\, t_i \in L_i\}.$$

6.11 Example Consider again $\Sigma = \{\alpha, \sigma\}$ from Example 5.5.
Let $L_1 = \{t \in \Sigma^{\circledast}[3,1] \mid \mathit{size}(t) \text{ is even}\}$, and $L_2 = \{t \in \Sigma^{\circledast}[3,1] \mid \mathit{size}(t) \text{ is odd}\}$. Then

$$L_1 \times L_2 = \{\langle t_1, t_2\rangle \in \Sigma^{\circledast}[3,2] \mid \mathit{size}(t_1) \text{ is even and } \mathit{size}(t_2) \text{ is odd}\}.$$

6.12 Proposition *Let Σ be a ranked alphabet and $n, m \in \mathbb{N}$. Further, let $L \subseteq \Sigma^{\circledast}[n,m]$. Then the following are equivalent.*

(i) $L \in \mathit{Rec}(\Sigma^{\circledast}, n, m)$.

(ii) There are $L_1, \ldots, L_m \in \mathit{Rec}(\Sigma^{\circledast}, n, 1)$, such that $L = L_1 \times \ldots \times L_m$.

Proof: For the first direction, let $L \in \mathit{Rec}(\Sigma^{\circledast}, n, m)$. Hence, there is an (n,m)-fsfa $\mathfrak{A} = \langle Q, \Sigma, I, F, E\rangle$ with $F = F_1 \times \ldots \times F_m$, such that $L(\mathfrak{A}) = L$. Now, for every $i = 1, \ldots, m$, construct the $(n,1)$-fsfa $\mathfrak{A}_i = \langle Q, \Sigma, I, F_i, E\rangle$. The following shows that $L(\mathfrak{A}) = L(\mathfrak{A}_1) \times \ldots \times L(\mathfrak{A}_m)$.

$$\begin{aligned}
t = \langle t_1, \ldots, t_m\rangle \in L(\mathfrak{A}) \quad &\iff \quad \exists \langle p_1, \ldots, p_m\rangle \in F \,.\, (\langle I_1, \ldots, I_m\rangle, t, \langle p_1, \ldots, p_m\rangle) \in E^{\mathfrak{A}}_{n,m} \\
&\iff \quad \forall i \in \{1, \ldots, m\} \,.\, \exists p_i \in F_i \,.\, (\langle I_1, \ldots, I_m\rangle, t_i, p_i) \in E^{\mathfrak{A}}_{n,1} \\
&\iff \quad \forall i \in \{1, \ldots, m\} \,.\, \exists p_i \in F_i \,.\, (\langle I_1, \ldots, I_m\rangle, t_i, p_i) \in E^{\mathfrak{A}_i}_{n,1} \\
&\iff \quad \forall i \in \{1, \ldots, m\} \,.\, t_i \in L(\mathfrak{A}_i)
\end{aligned}$$

Hence, the languages $L_1 = L(\mathfrak{A}_1), \dots, L_m = L(\mathfrak{A}_m)$ have the property needed in (ii).

Conversely, let $L_1, \dots, L_m \in Rec(\Sigma^{\circledast}, n, 1)$ be given such that $L = L_1 \times \dots \times L_m$. Hence, for every $i = 1, \dots, m$, there is an $(n, 1)$-dfsfa $\mathfrak{A}_i = \langle Q_i, \Sigma, \vec{q}_{I\,i}, F_i, \Delta_i \rangle$ with $L_i = L(\mathfrak{A}_i)$. For disambiguating the indices let $\vec{q}_{I\,i} = \langle q_{I\,i1}, \dots, q_{I\,in} \rangle$ and $\Delta_i = \{\delta_{ik} \mid k \geq 0\}$. Now construct an (n, m)-dfsfa $\mathfrak{A}' = \langle Q', \Sigma, \vec{q}_I{}', F', \Delta' \rangle$ with $L(\mathfrak{A}') = L = L_1 \times \dots \times L_m$. Before I go into details, let me informally sketch the idea of what is happening here. As mentioned before (see 5.15), an $(n, 1)$-dfsfa $\mathfrak{A} = \langle Q, \Sigma, \vec{q}_I, F, \Delta \rangle$ can be seen as a finite Σ-algebra with underlying set Q. The functions $\delta^{\mathfrak{A}}_{n,1}$ describe the unique homomorphism from the free Σ-algebra (to which $\Sigma^{\circledast}[n, 1]$ is isomorphic) into the finite Σ-algebra Q. The Σ-algebra described by $\mathfrak{A}'$ is simply the product of the algebras described by the $\mathfrak{A}_i$, and the unique homomorphism for $\mathfrak{A}'$ is the product of the homomorphisms for the $\mathfrak{A}_i$. With this in mind, it is an easy exercise to work out the details. Let

$$\begin{array}{rcl}
Q' &=& Q_1 \times \dots \times Q_m, \\
\vec{q}_I{}' &=& \langle q_{I}{}'_1, \dots, q_{I}{}'_n \rangle \\
&=& \langle \langle q_{I\,11}, \dots, q_{I\,m1} \rangle, \dots, \langle q_{I\,1n}, \dots, q_{I\,mn} \rangle \rangle, \\
F' &=& F'_1 \times \dots \times F'_m, \text{ where} \\
F'_1 &=& F_1 \times Q_2 \times \dots \times Q_m, \\
F'_2 &=& Q_1 \times F_2 \times Q_3 \times \dots \times Q_m, \\
&\vdots& \\
F'_m &=& Q_1 \times \dots \times Q_{m-1} \times F_m, \text{ and} \\
\Delta' &=& \{\delta'_k \mid k \in \mathbb{N}\}, \text{ where for every } k \geq 0, \\
\delta'_k(\langle \langle q_{11}, \dots, q_{m1} \rangle, \dots, \langle q_{1k}, \dots, q_{mk} \rangle \rangle, \sigma) &=& \langle \delta_{1k}(\langle q_{11}, \dots, q_{1k} \rangle, \sigma), \dots, \delta_{mk}(\langle q_{m1}, \dots, q_{mk} \rangle, \sigma) \rangle.
\end{array}$$

From this definition of the functions in Δ', it follows by an easy induction (a similar induction as for Claim 1 in the proof of Proposition 6.1) that for every $k \geq 0$, $t \in \Sigma^{\circledast}[k, 1]$, and $q_{11}, \dots, q_{1k} \in Q_1, \dots, q_{m1}, \dots, q_{mk} \in Q_m$,

$$\delta^{\mathfrak{A}'}_{k,1}(\langle \langle q_{11}, \dots, q_{m1} \rangle, \dots, \langle q_{1k}, \dots, q_{mk} \rangle \rangle, t) = \langle \delta^{\mathfrak{A}_1}_{k,1}(\langle q_{11}, \dots, q_{1k} \rangle, t), \dots, \delta^{\mathfrak{A}_m}_{k,1}(\langle q_{m1}, \dots, q_{mk} \rangle, t) \rangle.$$

It remains to show that indeed $L(\mathfrak{A}') = L$. Let $t = \langle t_1, \dots, t_m \rangle \in \Sigma^{\circledast}[n, m]$. Then

$$\begin{array}{rcl}
t \in L &\iff& t \in L_1 \times \dots \times L_m \\
&\iff& \forall i \in \{1, \dots, m\} \,.\, t_i \in L_i = L(\mathfrak{A}_i) \\
&\iff& \forall i \in \{1, \dots, m\} \,.\, \delta^{\mathfrak{A}_i}_{n,1}(\langle q_{I\,i1}, \dots, q_{I\,in} \rangle, t_i) \in F_i \\
&\iff& \forall i \in \{1, \dots, m\} \,.\, \delta^{\mathfrak{A}'}_{n,1}(\langle \langle q_{I\,11}, \dots, q_{I\,m1} \rangle, \dots, \langle q_{I\,1n}, \dots, q_{I\,mn} \rangle \rangle, t_i) \in F'_i \\
&\iff& \delta^{\mathfrak{A}'}_{n,m}(\langle q_{I}{}'_1, \dots, q_{I}{}'_n \rangle, \langle t_1, \dots, t_m \rangle) \in F'_1 \times \dots \times F'_m = F' \\
&\iff& t \in L(\mathfrak{A}')
\end{array}$$

Hence $L(\mathfrak{A}') = L$. □

6.13 Corollary *Let Σ be a ranked alphabet and $n \geq 0$ and $m \geq 1$ be natural numbers. Further, let $L \subseteq \Sigma^{\circledast}[n, m]$. If $L \in Rec(\Sigma^{\circledast}, n, m)$, then*

(i) $L; \pi^m_1, \dots, L; \pi^m_m \in Rec(\Sigma^{\circledast}, n, 1)$ *and*

(ii) $L = L; \pi^m_1 \times \dots \times L; \pi^m_m$.

Proof: Since $L \in Rec(\Sigma^{\circledast}, n, m)$, there are (by Proposition 6.12) $L_1, \dots, L_m \in Rec(\Sigma^{\circledast}, n, 1)$ such that $L = L_1 \times \dots \times L_m$. Hence, for every $i = 1, \dots, m$, we have $L; \pi^m_i = L_i \in Rec(\Sigma^{\circledast}, n, 1)$. Therefore, also

$L = L;\pi_1^m \times \ldots \times L;\pi_m^m$. $\square$

6.14 Example Now it is easy to see, why the language

$$L = L(\mathfrak{A}_a) \cup L(\mathfrak{A}_b) = \{\langle t_1, t_2\rangle \in \Sigma^{\circledast}[3,2] \mid \mathit{size}(\langle t_1, t_2\rangle) \text{ is odd}\}$$

from Example 6.6 is not recognizable. The reason is, that it is impossible to find languages $L_1, L_2 \in \mathit{Rec}(\Sigma^{\circledast}, n, 1)$ such that $L = L_1 \times L_2$. Whether a forest t_1 can occur in the first position of a pair $\langle t_1, t_2\rangle$ in L depends on its partner t_2. Therefore, it is not possible to describe L_1 independently from L_2.

6.15 This property of recognizable forest languages suggests a "union" operation for two languages $L, L' \subseteq \Sigma^{\circledast}[n,m]$ as follows. Whenever

$$\begin{aligned} L &= L_1 \times \ldots \times L_m \text{ for some } L_1, \ldots, L_m \subseteq \Sigma^{\circledast}[n,1] \text{ and} && (7)\\ L' &= L_1' \times \ldots \times L_m' \text{ for some } L_1', \ldots, L_m' \subseteq \Sigma^{\circledast}[n,1], && (8)\end{aligned}$$

then define

$$L \uplus L' = (L_1 \cup L_1') \times \ldots \times (L_m \cup L_m'). \qquad (9)$$

However, so far it is not clear, how this operation should be defined for languages that can not be written as cartesian products of $(n,1)$-forest languages (for some $n \in \mathbb{N}$). It could be any operation that satisfies (9) whenever L and L' have the form (7) and (8), respectively. But if L has not the form (7), then we can consider $\hat{L} = L;\pi_1^m \times \ldots \times L;\pi_m^m$, which certainly is a cartesian product of $(n,1)$-forest languages. This is the motivation for the following definition.

6.16 Definition Let $L, L' \subseteq \Sigma^{\circledast}[n,m]$ for some ranked alphabet Σ and two natural numbers n and m. Then

$$L \uplus L' = (L;\pi_1^m \cup L';\pi_1^m) \times \ldots \times (L;\pi_m^m \cup L';\pi_m^m)$$

is the *componentwise union* of L and L'.

6.17 Example Consider again the languages

$$\begin{aligned} L(\mathfrak{A}_a) &= \{\langle t_1, t_2\rangle \in \Sigma^{\circledast}[3,2] \mid \mathit{size}(t_1) \text{ is even and } \mathit{size}(t_2) \text{ is odd}\} \text{ and}\\ L(\mathfrak{A}_b) &= \{\langle t_1, t_2\rangle \in \Sigma^{\circledast}[3,2] \mid \mathit{size}(t_1) \text{ is odd and } \mathit{size}(t_2) \text{ is even}\}.\end{aligned}$$

from Example 6.6. Then

$$\begin{aligned} L(\mathfrak{A}_a) \uplus L(\mathfrak{A}_b) &= (\{t_1 \in \Sigma^{\circledast}[3,1] \mid \mathit{size}(t_1) \text{ is even}\} \cup \{t_1 \in \Sigma^{\circledast}[3,1] \mid \mathit{size}(t_1) \text{ is odd}\}) \times\\ &\quad\ (\{t_2 \in \Sigma^{\circledast}[3,1] \mid \mathit{size}(t_2) \text{ is odd}\} \cup \{t_2 \in \Sigma^{\circledast}[3,1] \mid \mathit{size}(t_2) \text{ is even}\})\\ &= \Sigma^{\circledast}[3,1] \times \Sigma^{\circledast}[3,1]\\ &= \Sigma^{\circledast}[3,2].\end{aligned}$$

6.18 For the sake of completeness, let me mention that the operation $\uplus$ has similar properties as its counterpart the usual set union $\cup$. It is associative and commutative. It can also be defined for an infinite number of languages as follows. Whenever there is a family $\{L_i \mid i \in I\}$ of languages indexed by an index set I such that $\forall i \in I \,.\, L_i \subseteq \Sigma^{\circledast}[n,m]$ (for some ranked alphabet Σ and two natural numbers n and m), then define

$$\biguplus_{i\in I} L_i \;=\; \left(\bigcup_{i\in I} L_i;\pi_1^m\right) \times \ldots \times \left(\bigcup_{i\in I} L_i;\pi_m^m\right).$$

6.19 Proposition *Let Σ be a ranked alphabet, and n,m be natural numbers. If $L, L' \in Rec(\Sigma^{\circledast}, n, m)$, then*

$$L \uplus L' \in Rec(\Sigma^{\circledast}, n, m).$$

Proof: Immediate from Corollary 6.13, Proposition 6.1 and Proposition 6.12. □

6.20 Corollary *For every ranked alphabet Σ and natural numbers n and m, the tuple*

$$\langle Rec(\Sigma^{\circledast}, n, m), \cap, \uplus, \emptyset, \Sigma^{\circledast}[n,m]\rangle$$

forms a boolean algebra.

Proof: This statement follows immediately from Corollary 6.4 and the well-known fact that the cartesian product of two boolean algebras is again a boolean algebra. Here, $Rec(\Sigma^{\circledast}, n, m)$ is the cartesian product of m copies of $Rec(\Sigma^{\circledast}, n, 1)$. □

6.21 The elements of the set $\Sigma^{\circledast}[n,m]$ (for a given ranked alphabet Σ and natural numbers $n \geq 0$ and $m \geq 1$) are tuples $t = \langle t_1, \ldots, t_m\rangle$. Each t_i can either represent a tree that contains at least one symbol from Σ, or it can be one of the empty trees $\pi_1^n, \ldots, \pi_n^n$. If we consider any (n,m)-fsfa $\mathfrak{A} = \langle Q, \Sigma, I, F, E\rangle$, then $L(\mathfrak{A})$ contains all tuples $\langle t_1, \ldots, t_m\rangle$ with $t_i = \pi_j^n$ (for some $i \in \{1,\ldots,m\}$ and $j \in \{1,\ldots,n\}$) if and only if $F_i \cap I_j \neq \emptyset$. This again emphasizes the fact that the empty trees π_j^n in the free polyoid play a similar role as the empty word in the free monoid. The set of all forests that contain an empty tree (i.e. $t_i = \pi_j^n$ for some $i \in \{1,\ldots,m\}$ and $j \in \{1,\ldots,n\}$) will play a special role in the remainder of this section, particularly in some of the proofs. I will denote this set by $\Pi(\Sigma^{\circledast}, n, m)$, which is for every ranked alphabet Σ and natural numbers n, m, formally defined as follows.

$$\Pi(\Sigma^{\circledast}, n, m) \;=\; \{\langle t_1, \ldots, t_m\rangle \in \Sigma^{\circledast}[n,m] \mid \exists i \in \{1,\ldots,m\} \,.\, \exists j \in \{1,\ldots,n\} \,.\, t_i = \pi_j^n\}.$$

6.22 Definition An (n,m)-fsfa $\mathfrak{A} = \langle Q, \Sigma, I, F, E\rangle$ is called *normalized* if

- $|I| = |F| = 1$ (i.e. there is exactly one input state tuple $\vec{q}_I = \langle q_{I1}, \ldots, q_{In}\rangle$ and exactly one output state tuple $\vec{q}_F = \langle q_{F1}, \ldots, q_{Fm}\rangle$), and
- for every $k \geq 0$, $\sigma \in \Sigma_k$, and $q_1, \ldots, q_k, p \in Q$, if $(\langle q_1, \ldots, q_k\rangle, \sigma, p) \in E_k$, then
  - $q_i \neq q_{Fj}$ for every $i = 1, \ldots, k$ and $j = 1, \ldots, m$, and
  - $p \neq q_{Il}$ for every $l = 1, \ldots, n$

  (i.e. there are no transitions from a final state or into an initial state).

6.23 Lemma *Let $\mathfrak{A}$ be an (n,m)-fsfa. Then there is a normalized (n,m)-fsfa $\mathfrak{A}'$ with*

$$L(\mathfrak{A}') \quad = \quad L(\mathfrak{A}) \setminus \Pi(\Sigma^{\circledast}, n, m). \tag{10}$$

Proof: Let $\mathfrak{A} = \langle Q, \Sigma, I, F, E\rangle$ with $I = I_1 \times \ldots \times I_n$ and $F = F_1 \times \ldots \times F_m$. Construct $\mathfrak{A}' = \langle Q', \Sigma, I', F', E'\rangle$ as follows. Choose $q_{I_1}, \ldots, q_{I_n}, q_{F_1}, \ldots, q_{F_m} \notin Q$ (all different from each other) and let

$$\begin{aligned}
Q' &= Q \cup \{q_{I_1}, \ldots, q_{I_n}, q_{F_1}, \ldots, q_{F_m}\} \\
I' &= \{\langle q_{I_1}, \ldots, q_{I_n}\rangle\} \\
F' &= \{\langle q_{F_1}, \ldots, q_{F_m}\rangle\} \\
E'_k &= E_k \;\cup \bigcup_{K \subseteq \{1,\ldots,k\}} E^1_{k,K} \;\cup \bigcup_{l \in \{1,\ldots,m\}} E^2_{k,l} \;\cup \bigcup_{\substack{K \subseteq \{1,\ldots,k\} \\ l \in \{1,\ldots,m\}}} E^3_{k,K,l} \quad \text{where}
\end{aligned}$$

$$\begin{aligned}
E^1_{k,K} = \{\ (\langle p_1, \ldots, p_k\rangle, \sigma, q) \ \mid\ & \exists(\langle q_1, \ldots, q_k\rangle, \sigma, q) \in E_k \,.\, \forall i \in \{1, \ldots, k\} \,. \\
& (i \notin K \wedge p_i = q_i) \vee (i \in K \wedge \exists j \in \{1, \ldots, n\} \,.\, (q_i \in I_j \wedge p_i = q_{I_j}))\ \}
\end{aligned}$$

$$E^2_{k,l} = \{\ (\langle q_1, \ldots, q_k\rangle, \sigma, q_{F_l}) \ \mid\ \exists(\langle q_1, \ldots, q_k\rangle, \sigma, q) \in E_k \,.\, q \in F_l\ \}$$

$$\begin{aligned}
E^3_{k,K,l} = \{\ (\langle p_1, \ldots, p_k\rangle, \sigma, q_{F_l}) \ \mid\ & \exists(\langle q_1, \ldots, q_k\rangle, \sigma, q) \in E_k \,.\, q \in F_l \wedge \forall i \in \{1, \ldots, k\} \,. \\
& (i \notin K \wedge p_i = q_i) \vee (i \in K \wedge \exists j \in \{1, \ldots, n\} \,.\, (q_i \in I_j \wedge p_i = q_{I_j}))\ \}
\end{aligned}$$

Informally speaking, for every $k \geq 0$ and every state transition in E_k that goes into a final state $q \in F_l$ (for some $l = 1, \ldots, m$), a new state transition that goes into q_{F_l} is put into E'_k (via $E^2_{k,l}$). For every state transition in E_k that comes from a state tuple $\langle p_1, \ldots, p_k\rangle$ that contains some initial states, a new state transition, in which every initial state $q_i \in I_j$ is replaced by q_{I_j}, is put into E'_k (via $E^1_{k,K}$). Further, for every state transition in E_k that comes from some initial state $q_i \in I_j$ and goes into a final state $q \in F_l$, a new state transition where q_i and q are replaced by q_{I_j} and q_{F_l} (respectively) is put into E'_k (via $E^3_{k,K,l}$).
It is easy to see that $\mathfrak{A}'$ is normalized, because there are no state transitions from a final state q_{F_l} (for any $l = 1, \ldots, m$) or into an initial state q_{I_j} (for any $j = 1, \ldots, n$). It remains to show that $L(\mathfrak{A}')$ fulfills (10). Let $t \in \Sigma^{\circledast}[n,m]$ such that $t \notin \Pi(\Sigma^{\circledast}, n, m)$. Then

$$\begin{aligned}
t \in L(\mathfrak{A}) \quad &\Longleftrightarrow \quad \exists\langle p_1, \ldots, p_m\rangle \in F \,.\, (\langle I_1, \ldots, I_n\rangle, t, \langle p_1, \ldots, p_m\rangle) \in E^{\mathfrak{A}}_{n,m} \\
&\qquad\quad \text{(by Definition 5.4)} \\
&\Longleftrightarrow \quad \exists\langle p_1, \ldots, p_m\rangle \in F \,.\, (\langle I_1, \ldots, I_n\rangle, t, \langle p_1, \ldots, p_m\rangle) \in E^{\mathfrak{A}'}_{n,m} \\
&\qquad\quad \text{(since for every } k \geq 0,\ E_k \subseteq E'_k, \text{ and } q_{I_1}, \ldots, q_{I_n}, q_{F_1}, \ldots, q_{F_m} \notin Q) \\
&\Longleftrightarrow \quad (\langle \{q_{I_1}\}, \ldots, \{q_{I_n}\}\rangle, t, \langle q_{F_1}, \ldots, q_{F_m}\rangle) \in E^{\mathfrak{A}'}_{n,m} \\
&\qquad\quad \text{(by the construction of } E^1_{k,K},\ E^2_{k,l}, \text{ and } E^3_{k,K,l}, \text{ and by the fact that } t \notin \Pi(\Sigma^{\circledast}, n, m)) \\
&\Longleftrightarrow \quad t \in L(\mathfrak{A}') \text{ (by Definition 5.4)}
\end{aligned}$$

Since $\{q_{I_1}, \ldots, q_{I_n}\} \cap \{q_{F_1}, \ldots, q_{F_m}\} = \emptyset$, we have $(\langle \{q_{I_1}\}, \ldots, \{q_{I_n}\}\rangle, \pi^n_i, q_{F_j}) \notin E^{\mathfrak{A}'}_{n,1}$ for every $i = 1, \ldots, n$ and $j = 1, \ldots, m$. Hence,

$$L(\mathfrak{A}') \cap \Pi(\Sigma^{\circledast}, n, m) = \emptyset.$$

Therefore, $L(\mathfrak{A}') = L(\mathfrak{A}) \setminus \Pi(\Sigma^{\circledast}, n, m)$. □

6.24 The concatenation $L_1; L_2$ of two languages $L_1 \subseteq \Sigma^{\circledast}[n,m]$ and $L_2 \subseteq \Sigma^{\circledast}[m,k]$, as it is defined in Definition 6.7, seems to be natural. However, this naive concatenation has two serious problems that cause this operation to fail in being the right basis for a star operation that is necessary for the main theorem in the next section. It is obvious that

$$L_1; L_2 = \bigcup_{t_2 \in L_2} L_1; t_2.$$

The first problem now arises because the recognizable sets are not closed under union. It is possible to circumvent this problem by using the operation $\biguplus$ that has been defined in 6.18, but there is a second problem. Consider, for example, the set $L; t$ for some $L \subseteq \Sigma^{\circledast}[0, 2]$ (where $\Sigma = \{\alpha, \sigma\}$ is taken from Example 5.5) and $t = \langle \pi_1^2, \pi_1^2 \rangle \sigma \in \Sigma^{\circledast}[2, 1]$. Then

$$L; t = \{\langle s, s \rangle \sigma \mid s \in L; \pi_1^2\} \subseteq \Sigma^{\circledast}[0, 1].$$

Observe that the same forest s is plugged into the first as well as the second argument position of σ. However, in order to get an equivalence between recognizable and rational forest languages in the next section, the concatenation of L and t must deliver the set $\{\langle s, s' \rangle \sigma \mid s, s' \in L; \pi_1^2\}$, where both argument positions of σ can be filled with two different trees s and s' from $L; \pi_1^2$. The following definitions provide the necessary technicalities.

6.25 Definition Let Σ be a ranked alphabet and $n, m, k \in \mathbb{N}$. Further, let $L \subseteq \Sigma^{\circledast}[n, m]$ and $t \in \Sigma^{\circledast}[m, k]$. Define the *concatenation* $L \cdot t$ of L and t by induction on t as follows.

$$\begin{array}{rcl}
L \cdot \langle\rangle_m & = & \{\langle\rangle_n\} \\
L \cdot \pi_i^m & = & L; \pi_i^m \quad \text{(for every } i = 1, \ldots, m) \\
L \cdot (t'\sigma) & = & (L \cdot t'); \varepsilon_{k'}\sigma \quad \text{(for every } k' \geq 0,\ t' \in \Sigma^{\circledast}[m, k'] \text{, and } \sigma \in \Sigma_{k'}) \\
L \cdot \langle t_1, \ldots, t_k \rangle & = & L \cdot t_1 \times \ldots \times L \cdot t_k \quad \text{(for every } t_1, \ldots, t_k \in \Sigma^{\circledast}[m, 1])
\end{array}$$

6.26 Remark The construction of the forest concatenation might seem unnatural, but the usual tree concatenation (cf. [TW68] or [GS97]) is defined in a similar way: Different occurrences of the same 0-ary symbol can be replaced by different trees.

6.27 Definition Let Σ be a ranked alphabet and $n, m, k \in \mathbb{N}$. Further, let $L_1 \subseteq \Sigma^{\circledast}[n, m]$ and $L_2 \subseteq \Sigma^{\circledast}[m, k]$. Then

$$L_1 \cdot L_2 = \biguplus_{t_2 \in L_2} L_1 \cdot t_2$$

is the *concatenation* of L_1 and L_2.

6.28 Lemma *Let Σ be a ranked alphabet and $n, m, k \in \mathbb{N}$. Further, let $L_1 \subseteq \Sigma^{\circledast}[n, m]$ and $L_2, L_3 \subseteq \Sigma^{\circledast}[m, k]$. Then*

$$L_1 \cdot (L_2 \uplus L_3) = (L_1 \cdot L_2) \uplus (L_1 \cdot L_3).$$

Proof: Immediate from Definition 6.27.

$$L_1 \cdot (L_2 \uplus L_3) = \biguplus_{t \in (L_2 \uplus L_3)} L_1 \cdot t = \biguplus_{t \in L_2} L_1 \cdot t \uplus \biguplus_{t \in L_3} L_1 \cdot t = (L_1 \cdot L_2) \uplus (L_1 \cdot L_3).$$

□

6.29 Lemma *Let* Σ *be a ranked alphabet and* $n, m, k \in \mathbb{N}$. *Further, let* $L \subseteq \Sigma^{\circledast}[n,m]$ *and* $L \subseteq \Sigma^{\circledast}[m,k]$. *If* $L' = L'_1 \times \ldots \times L'_k$ *for some* $L'_1, \ldots, L'_k \subseteq \Sigma^{\circledast}[m,1]$, *then*

$$L \cdot L' \;=\; L \cdot L'_1 \times \ldots \times L \cdot L'_k.$$

Proof: The result follows immediately from Definitions 6.27, 6.25, and 6.16:

$$\begin{aligned}
L \cdot L' &= \biguplus_{t' \in L'} L \cdot t' \\
&= \biguplus_{\langle t'_1,\ldots,t'_k\rangle \in L'} (L \cdot t'_1 \times \ldots \times L \cdot t'_k) \\
&= \left(\bigcup_{t'_1 \in L'_1} L \cdot t'_1\right) \times \ldots \times \left(\bigcup_{t'_k \in L'_k} L \cdot t'_k\right) \\
&= \left(\biguplus_{t'_1 \in L'_1} L \cdot t'_1\right) \times \ldots \times \left(\biguplus_{t'_k \in L'_k} L \cdot t'_k\right) \\
&= L \cdot L'_1 \times \ldots \times L \cdot L'_k
\end{aligned}$$

$\square$

6.30 Remark Observe that in Lemma 6.29, the language L can be any language in $\Sigma^{\circledast}[n,m]$, whereas $L' \subseteq \Sigma^{\circledast}[m,k]$ has to be of the form (8).

6.31 Lemma *Let* Σ *be a ranked alphabet and* $n, m, k \in \mathbb{N}$. *Further, let* $L \subseteq \Sigma^{\circledast}[n,m]$, $L' \subseteq \Sigma^{\circledast}[m,k]$, *and* $\sigma \in \Sigma_k$. *If* $L' = L'_1 \times \ldots \times L'_k$ *for some* $L'_1, \ldots, L'_k \subseteq \Sigma^{\circledast}[m,1]$, *then*

(i) $L' \cdot \varepsilon_k\sigma = L'; \varepsilon_k\sigma = \{\langle t_1, \ldots, t_k\rangle\sigma \in \Sigma^{\circledast}[m,1] \mid \forall i \in \{1,\ldots,k\} \;.\; t_i \in L'_i\}$ *and*

(ii) $(L \cdot L'); \varepsilon_k\sigma = L \cdot (L'; \varepsilon_k\sigma)$.

Proof: (i) Immediate from Definition 6.25. (ii) First, we have $(L \cdot L'); \varepsilon_k\sigma = \left(\biguplus_{t' \in L'} L \cdot t'\right); \varepsilon_k\sigma$. Since $L' = L'_1 \times \ldots \times L'_k \subseteq \Sigma^{\circledast}[m,k]$, this is equal to $\biguplus_{t' \in L'} L \cdot t'\sigma$ $(= \bigcup_{t' \in L'} L \cdot t'\sigma)$. From this, the statement follows because

$$\biguplus_{t' \in L'} L \cdot t'\sigma \;=\; \biguplus_{t' \in L'} L \cdot (t'; \varepsilon_k\sigma) \;=\; \biguplus_{t'' \in L'; \varepsilon_k\sigma} L \cdot t'' \;=\; L \cdot (L'; \varepsilon_k\sigma).$$

$\square$

6.32 Lemma *Let* Σ *be a ranked alphabet and* $n, m, k, l \in \mathbb{N}$. *Further, let* $L \subseteq \Sigma^{\circledast}[n,m]$, $L' \subseteq \Sigma^{\circledast}[m,k]$, *and* $t \in \Sigma^{\circledast}[k,l]$. *If* $L' = L'_1 \times \ldots \times L'_k$ *for some* $L'_1, \ldots, L_k \subseteq \Sigma^{\circledast}[m,k]$, *then*

$$(L \cdot L') \cdot t \;=\; L \cdot (L' \cdot t).$$

Proof: The proof (done by case analysis on l and induction on the structure of t) will be very similar to the proof of Lemma 4.33.

- $\underline{l = 0.}$ Then $t = \langle\rangle_k$. Therefore

$$(L \cdot L') \cdot t = (L \cdot L') \cdot \langle\rangle_k = \{\langle\rangle_n\} = L \cdot \langle\rangle_m = L \cdot \{\langle\rangle_m\} = L \cdot (L' \cdot \langle\rangle_k) = L \cdot (L' \cdot t).$$

- $\underline{l = 1.}$ Then there are two cases:
  - $\underline{t = \pi_i^k}$ for some $i = 1, \ldots, k$. Then

    $$(L \cdot L') \cdot t = (L \cdot (L'_1 \times \ldots \times L'_k)) \cdot \pi_i^k = (L \cdot L'_1 \times \ldots \times L \cdot L'_k) \cdot \pi_i^k = L \cdot L'_i = L \cdot (L' \cdot \pi_i^k) = L \cdot (L' \cdot t).$$

  - $\underline{t = t'\sigma}$ for some $k' \in \mathbb{N}$, some $\Sigma \in \Sigma_{k'}$, and some $t' \in \Sigma^{\circledast}[k, k']$. Then

    $$\begin{aligned}(L \cdot L') \cdot t &= (L \cdot L') \cdot t'\sigma \\ &= ((L \cdot L') \cdot t'); \varepsilon_k \sigma \quad \text{(by Definition 6.25)} \\ &= (L \cdot (L' \cdot t')); \varepsilon_k \sigma \quad \text{(by induction hypothesis)} \\ &= L \cdot ((L' \cdot t'); \varepsilon_k \sigma) \quad \text{(by Lemma 6.31)} \\ &= L \cdot (L' \cdot t'\sigma) \quad \text{(by Definition 6.25)} \\ &= L \cdot (L' \cdot t).\end{aligned}$$

- $\underline{l \geq 2.}$ Then $t = \langle t_1, \ldots, t_l \rangle$ for some $t_1, \ldots, t_l \in \Sigma^{\circledast}[k, 1]$. Therefore

  $$\begin{aligned}(L \cdot L') \cdot t &= (L \cdot L') \cdot \langle t_1, \ldots, t_l \rangle \\ &= ((L \cdot L') \cdot t_1) \times \ldots \times ((L \cdot L') \cdot t_l) \quad \text{(by Definition 6.25)} \\ &= (L \cdot (L' \cdot t_1)) \times \ldots \times (L \cdot (L' \cdot t_l)) \quad \text{(by induction hypothesis)} \\ &= L \cdot (L' \cdot t_1 \times \ldots \times L' \cdot t_l) \quad \text{(by Lemma 6.29)} \\ &= L \cdot (L' \cdot \langle t_1, \ldots, t_l \rangle) \quad \text{(by Definition 6.25)} \\ &= L \cdot (L' \cdot t).\end{aligned}$$

□

6.33 Proposition *Let Σ be a ranked alphabet, let $n, m, k, l \in \mathbb{N}$, and let $L =\subseteq \Sigma^{\circledast}[n, m]$, $L' \subseteq \Sigma^{\circledast}[m, k]$, and $L'' \subseteq \Sigma^{\circledast}[k, l]$. If $L' = L'_1 \times \ldots \times L'_k$ for some $L'_1, \ldots, L'_k \subseteq \Sigma^{\circledast}[m, 1]$ and $L'' = L''_1 \times \ldots \times L''_l$ for some $L''_1, \ldots, L''_l \subseteq \Sigma^{\circledast}[k, 1]$, then*

$$(L \cdot L') \cdot L'' \ = \ L \cdot (L' \cdot L'').$$

Proof:

$$\begin{aligned}(L \cdot L') \cdot L'' &= \biguplus_{t'' \in L''} (L \cdot L') \cdot t'' \quad \text{(by Definition 6.27)} \\ &= \biguplus_{t'' \in L''} L \cdot (L' \cdot t'') \quad \text{(by Lemma 6.32)} \\ &= L \cdot \left(\biguplus_{t'' \in L''} L' \cdot t'' \right) \quad \text{(by Lemma 6.28 and since } L'' = L''_1 \times \ldots \times L''_l) \\ &= L \cdot (L' \cdot L'') \quad \text{(by Definition 6.27)}\end{aligned}$$

□

6.34 Proposition *Let Σ be a ranked alphabet, and n, m, k be natural numbers. If $L_1 \in Rec(\Sigma^{\circledast}, n, m)$ and $L_2 \in Rec(\Sigma^{\circledast}, m, k)$, then*

$$L_1 \cdot L_2 \in Rec(\Sigma^{\circledast}, n, k).$$

Proof: Let $\mathfrak{A}_1 = \langle Q_1, \Sigma, I_1, F_1, E_1\rangle$ be an (n,m)-fsfa and $\mathfrak{A}_2 = \langle Q_2, \Sigma, I_2, F_2, E_2\rangle$ be an (m,k)-fsfa with

$$L(\mathfrak{A}_1) = L_1 \quad \text{and} \quad L(\mathfrak{A}_2) = L_2. \tag{11}$$

Construct the normalized automata $\mathfrak{A}_1' = \langle Q_1', \Sigma, I_1', F_1', E_1'\rangle$ and $\mathfrak{A}_2' = \langle Q_2', \Sigma, I_2', F_2', E_2'\rangle$ by Lemma 6.23. Now, the (n,k)-automaton $\mathfrak{A}'' = \langle Q'', \Sigma, I'', F'', E''\rangle$ is obtained by concatenating $\mathfrak{A}_1'$ and $\mathfrak{A}_2'$ such that $F_1' = \{\langle q_{F11}, \ldots, q_{F1m}\rangle\}$ and $I_2' = \{\langle q_{I21}, \ldots, q_{I2m}\rangle\}$ are identified. Formally this is done as follows:

$$\begin{aligned}
Q'' &= Q_2' \cup Q_1' \setminus \{q_{F11}, \ldots, q_{F1m}\}\\
E_k'' &= E_{2k}' \cup \{(\langle q_1, \ldots, q_k\rangle, \sigma, p) \mid \exists(\langle q_1, \ldots, q_k\rangle, \sigma, q) \in E_{1k}' \,.\\
&\quad (\exists i \in \{1, \ldots, m\} \,.\, q = q_{F1i} \wedge p = q_{I2i}) \vee (\forall i \in \{1, \ldots, m\} \,.\, q \neq q_{F1i} \wedge p = q)\}
\end{aligned}$$

For the definition of $I'' = I_1'' \times \ldots \times I_n''$ and $F'' = F_1'' \times \ldots \times F_k''$, we cannot simply take I_1' and F_2' because $L(\mathfrak{A}_1') \neq L_1$ and $L(\mathfrak{A}_2') \neq L_2$ in the general case.
In order to simplify the following equations, let me assume that $L_1 = L_{11} \times \ldots \times L_{1m}$, where $L_{11}, \ldots, L_{1m} \in Rec(\Sigma^{\circledast}, n, 1)$ and $L_2 = L_{21} \times \ldots \times L_{2k}$, where $L_{21}, \ldots, L_{2k} \in Rec(\Sigma^{\circledast}, m, 1)$, which is possible because of Proposition 6.12. Now, define for every $i = 1, \ldots, n$, the set I_i'' and for every $j = 1, \ldots, k$, the set F_j'' as follows

$$I_i'' = \{q_{I1i}\} \cup \{\, q_{I2l} \mid \pi_i^n \in L_{1l} \,\} \tag{12}$$
$$F_j'' = \{q_{F2j}\} \cup \{\, q_{I2l} \mid \pi_l^m \in L_{2j} \,\} \tag{13}$$

Now it remains to show, that indeed $L(\mathfrak{A}'') = L_1 \cdot L_2$.
First, observe that $L(\mathfrak{A}'') = L(\mathfrak{A}''); \pi_1^k \times \ldots \times L(\mathfrak{A}''); \pi_k^k$ (by Corollary 6.13). Furthermore, $L_1 \cdot L_2 = L_1 \cdot L_{21} \times \ldots \times L_1 \cdot L_{2k}$ by Lemma 6.29. Therefore, it suffices to show that $L(\mathfrak{A}''); \pi_j^k = L_1 \cdot L_{2j}$ for every $j = 1, \ldots, k$.

Suppose $t \in L(\mathfrak{A}''); \pi_j^k$. Then there is a $p_j \in F_j''$ such that $(\langle I_1'', \ldots, I_n''\rangle, t, p_j) \in E_{n,1}^{\mathfrak{A}''}$. By (13), there are two cases to be distinguished.

- $\underline{p_j = q_{F2j}}$. Then there are languages $L_{11}', \ldots, L_{1m}' \subseteq \Sigma^{\circledast}[n,1]$ and an $s \in L(\mathfrak{A}_2'); \pi_j^k$ such that $\underline{t \in (L_{11}' \times \ldots \times L_{1m}') \cdot s}$. To see what the languages $L_{11}', \ldots, L_{1m}'$ are, consider an $r \in \Sigma^{\circledast}[n,1]$. Then, for every $l = 1, \ldots, m$,

$$\begin{aligned}
r \in L_{1l}' &\iff (\exists i \in \{1, \ldots, n\}. r = \pi_i^n \wedge \pi_i^n \in L_{1l}) \vee r \in L(\mathfrak{A}_1'); \pi_l^m \quad \text{(by (12))}\\
&\iff r \in L_{1l} \;\text{ because, by (10), } L(\mathfrak{A}_1); \pi_l^m = L_{1l} \setminus \{\pi_1^n, \ldots, \pi_n^n\}
\end{aligned}$$

  Hence, $L_{1l}' = L_{1l}$. And therefore, $t \in L_1 \cdot s$. Since $L(\mathfrak{A}_2'); \pi_j^k \subseteq L_{2j}$, this means that $t \in L_1 \cdot L_{2j}$.
- $\underline{p_j = q_{I2l} (= q_{F1l})}$ for some $l = 1, \ldots, m$. Then $\pi_l^m \in L_{2j}$. Furthermore, either $t \in L(\mathfrak{A}_1'); \pi_l^m$ or $\underline{t = \pi_i^n}$ for some $i = 1, \ldots, n$ with $\pi_i^n \in L_{1l}$ (by (12)). In both cases $t \in L_1 \cdot L_{2j}$.

For the other direction, let $t \in L_1 \cdot L_{2j}$. Hence, there is an $s \in L_{2j}$ with $t \in L_1 \cdot s$. Again, there are two cases.

- $\underline{s = \pi_l^m}$ for some $l = 1, \ldots, m$. Then $q_{I2l} = q_{F1l} \in F_j''$ by (13). Furthermore, $t \in L_{1l} = L(\mathfrak{A}_1); \pi_l^m$ (by Corollary 6.13). There are again two cases. Either $t = \pi_i^n$ for some $i = 1, \ldots, n$, or $t \neq \pi_i^n$ for all $i = 1, \ldots, n$. In the first case $q_{I2l} = q_{F1l} \in I_i''$ by (12), and in the second case $t \in L(\mathfrak{A}_1'); \pi_l^m$ by (10). In both cases $t \in L(\mathfrak{A}''); \pi_j^k$.
- $\underline{s \neq \pi_l^m}$ for all $l = 1, \ldots, m$. Then $s \in L(\mathfrak{A}_2'); \pi_j^k$. By way of contradiction, assume $t \notin L(\mathfrak{A}''); \pi_j^k$. Then there must be an $l \in \{1, \ldots, m\}$ and an $r \in \Sigma^{\circledast}[n,1]$ such that $r \in L_{1l}$ (because $t \in L_1 \cdot s$) and (since $t \notin L(\mathfrak{A}''); \pi_j^k$) $r \notin L(\mathfrak{A}_1'); \pi_l^m$ and (by (12)) $r \neq \pi_i^n$ for every $i = 1, \ldots, n$ with $\pi_i^n \in L1j$. By (10), this is impossible.

Hence, $L(\mathfrak{A}'') = L_1 \cdot L_2$. $\square$

6.35 At this point, all the spadework has been done, that was necessary to define a generalization of Kleene's star-operation. The leading idea behind the star is to be a closure operation for the concatenation. For the concatenation operation of a free monoid this does not bring any problems. However, for the concatenation of forest languages (as defined in Definition 6.27), this makes sense only for languages $L \subseteq \Sigma^{\circledast}[n,n]$ for some ranked alphabet Σ and $n \in \mathbb{N}$. The next problem is, that we cannot just take the union of all L^z for $0 \leq z < \omega$ as it is done for word languages because the recognizable forest languages are not closed under union. However, there is the operation $\biguplus$, which will solve this problem.

6.36 Definition Let $L \subseteq \Sigma^{\circledast}[n,n]$ for some ranked alphabet Σ and some natural number n such that $L \neq \emptyset$. Then define

$$\begin{aligned} L^0 &= \{\langle \pi_1^n, \ldots, \pi_n^n \rangle\} \;=\; \{\varepsilon_n\}, \\ L^{z+1} &= L^z \cdot L \quad \text{for every } z \geq 0, \text{ and} \\ L^\star &= \biguplus_{z \geq 0} L^z. \end{aligned}$$

6.37 I excluded the empty set because for every $n \in \mathbb{N}$, we have $\emptyset \subseteq \Sigma^{\circledast}[n,n]$. But we also have for every $n, m \in \mathbb{N}$ and $L \subseteq \Sigma^{\circledast}[n,m]$ that $\emptyset \cdot L = \emptyset = L \cdot \emptyset$ and $\emptyset \uplus L = L = L \uplus \emptyset$. Hence, by Definition 6.36, we would get $\emptyset^\star = \emptyset^0$, but it is not clear, what $\emptyset^0$ should be. In order to avoid complicated case analyses in some later proofs, let me use the following convention. Whenever the discussion is casted in terms of a language $L \subseteq \Sigma^{\circledast}[n,n]$ for some ranked alphabet Σ and some natural number n, and this language happens to be the empty set, then by definition put $L^\star = L^0 = \{\varepsilon_n\}$.

6.38 Remark Observe that this star operation is a generalization of Kleene's star operation for word languages as well as a generalization of the tree concatenation closure operations defined in [TW68].

6.39 Lemma *Let $L \subseteq \Sigma^{\circledast}[n,n]$ for some ranked alphabet Σ and some natural number n. Then $L^\star = L_1 \times \ldots \times L_n$ for some $L_1, \ldots, L_n \subseteq \Sigma^{\circledast}[n,1]$.*

Proof: Immediate from the definition of $\biguplus$ (see 6.18). $\square$

6.40 Proposition *Let Σ be a ranked alphabet, and n be a natural number. If $L \in Rec(\Sigma^{\circledast}, n, n)$, then*

$$L^\star \in Rec(\Sigma^{\circledast}, n, n).$$

Proof: Since $L \in Rec(\Sigma^{\circledast}, n, n)$, we have (by Proposition 6.12) that $L = L_1 \times \ldots \times L_n$ for some $L_1, \ldots, L_n \in Rec(\Sigma^{\circledast}, n, 1)$. Let $\mathfrak{A} = \langle Q, \Sigma, I, F, E \rangle$ be an (n,n)-fsfa such that $L(\mathfrak{A}) = L$. Construct the normalized automaton $\mathfrak{A}' = \langle Q', \Sigma, I', F', E' \rangle$ (by Lemma 6.23) with $F' = \{\langle q_{F1}, \ldots, q_{Fn} \rangle\}$ and $I' = \{\langle q_{I1}, \ldots, q_{In} \rangle\}$. Now construct $\mathfrak{A}'' = \langle Q'', \Sigma, I'', F'', E'' \rangle$ by identifying I' and F'. Observer that this does not suffice because $L(\mathfrak{A}'')$ would be $L(\mathfrak{A}')^\star$, which is not desired because $L(\mathfrak{A}') \neq L$. Therefore the set of final state tuples has to be modified in a similar way as in the proof of Proposition 6.34.

Formally, the whole construction of $\mathfrak{A}''$ spells out as follows.

$$\begin{array}{rcl}
Q'' & = & Q' \setminus \{q_{F1}, \ldots, q_{Fn}\} \\
I'' & = & I' \\
F'' & = & F_1'' \times \ldots \times F_n'', \text{ where} \\
F_i'' & = & \{q_{Ii}\} \cup \{\, q_{Ij} \mid \pi_j^n \in L_i \,\} \quad (\text{for every } i = 1, \ldots, n) \\
E_k'' & = & \{(\langle p_1, \ldots, p_k\rangle, \sigma, p_{k+1}) \mid \exists(\langle q_1, \ldots, q_k\rangle, \sigma, q_{k+1}) \in E_k' \,.\, \forall j \in \{1, \ldots, k+1\} \,. \\
 & & \qquad (\exists i \in \{1, \ldots, n\} \,.\, q_j = q_{Fi} \wedge p_j = q_{Ii}) \vee (\forall i \in \{1, \ldots, n\} \,.\, q_j \neq q_{Fi} \wedge p_j = q_j)\}
\end{array}$$

In other words, for every $k \geq 0$, the relation E_k'' is obtained from E_k' by replacing q_{Fi} by q_{Ii} for every $i = 1, \ldots, n$.
I will now show that $L(\mathfrak{A}'') = L^\star$ by showing that $L(\mathfrak{A}''); \pi_i^n = L^\star; \pi_i^n$ for every $i = 1, \ldots, n$.

- First, I will show by induction on z that for every $i = 1, \ldots, n$,

$$\left(\biguplus_{\zeta \leq z} L^\zeta\right); \pi_i^n \subseteq L(\mathfrak{A}''); \pi_i^n.$$

  - Induction Base ($\zeta = 0$): Let $t \in \left(\biguplus_{\zeta \leq 0} L^\zeta\right); \pi_i^n$. Then $t \in L^0; \pi_i^n = \{\pi_i^n\}$. Hence, $t = \pi_i^n$, and therefore $t \in L(\mathfrak{A}''); \pi_i^n$ (since $I_i'' \subseteq F_i''$).
  - Induction Step: Let $t \in \left(\biguplus_{\zeta \leq z+1} L^\zeta\right); \pi_i^n$. Then $t \in \left(\biguplus_{\zeta \leq z} L^\zeta\right); \pi_i^n$ or $t \in \left(\biguplus_{\zeta \leq z} L^\zeta\right) \cdot L_i$. In the first case, $t \in L(\mathfrak{A}''); \pi_i^n$ by induction hypothesis. In the second case, there is an $s \in L_i$ such that $t \in \left(\biguplus_{\zeta \leq z} L^\zeta\right) \cdot s$, and therefore (by induction hypothesis) $t \in L(\mathfrak{A}'') \cdot s$.
    - If $s = \pi_j^n$ for some $j = 1, \ldots, n$, then $q_{Ij} \in F_i''$. Therefore $t \in L(\mathfrak{A}''); \pi_i^n$.
    - If $s \neq \pi_j^n$ for every $j = 1, \ldots, n$, then $s \in L(\mathfrak{A}'); \pi_i^n$. Therefore $t \in L(\mathfrak{A}''); \pi_i^n$.

  This implies that $L^\star; \pi_i^n \subseteq L(\mathfrak{A}''); \pi_i^n$ for every $i = 1, \ldots, n$. Therefore $L^\star \subseteq L(\mathfrak{A}'')$.

- For the other direction let $t \in L(\mathfrak{A}'')$. By way of contradiction assume that $t \notin L^\star$.
  Since $t \in L(\mathfrak{A}'')$, we have $t = \langle t_1, \ldots, t_n\rangle$ with $t_i \in L(\mathfrak{A}''); \pi_i^n$ for every $i = 1, \ldots, n$. Hence, for every $i = 1, \ldots, n$ there is a $p_i \in F_i''$ such that

$$(\langle\{q_{I1}\}, \ldots, \{q_{In}\}\rangle, t_i, p_i) \in E_{n,1}^{\mathfrak{A}''}.$$

  Since $t \notin L^\star$, there is an $i = 1, \ldots, n$ such that $t_i \notin L^\star; \pi_i^n$. By the definition of F'', there are two cases.

  - $p_i = q_{Ii}$. Then there are again two cases.
    - $t_i = \pi_i^n$. Then $t_i \in L^0; \pi_i^n$ and therefore $t_i \in L^\star; \pi_i^n$ which is a contradiction.
    - $t_i \in L(\mathfrak{A}'') \cdot s$ for some $s \in L(\mathfrak{A}'); \pi_i^n$. Then $t_i \in L(\mathfrak{A}'') \cdot L_i$ because $L(\mathfrak{A}'); \pi_i^n \subseteq L_i$. Since $t_i \notin L^\star; \pi_i^n$ and $\mathit{size}(s) > 0$, there is a $t' \in \Sigma^\circledast[n, 1]$ with $\mathit{size}(t') < \mathit{size}(t_i)$ and an $i' \in \{1, \ldots, n\}$ such that $t' \in L(\mathfrak{A}''); \pi_{i'}^n$ and $t' \notin L^\star; \pi_{i'}^n$. Since $\mathit{size}(t_i)$ is finite, this will lead (by an inductive argument on the size of t_i) to a contradiction eventually.
  - $p_i = q_{Ij}$ for some $j = 1, \ldots, n$ with $\pi_j^n \in L_i$. Then (again) there are two cases.
    - $t_i = \pi_j^n$. Then $t_i \in L_i = L^1; \pi_i^n$ and therefore $t_i \in L^\star; \pi_i^n$ which is a contradiction.
    - $t_i \in L(\mathfrak{A}'') \cdot s$ for some $s \in L(\mathfrak{A}'); \pi_j^n$. Then $t_i \in L(\mathfrak{A}'') \cdot (L \cdot L_i)$ because $L(\mathfrak{A}') \subseteq L$ and $\pi_j^n \in L_i$. It follows the same argumentation as above. Since $t_i \notin L^\star; \pi_i^n$, there is an $i' \in \{1, \ldots, n\}$ and a $t' \in L(\mathfrak{A}''); \pi_{i'}^n$ with $t' \notin L^\star; \pi_{i'}^n$. Since $\mathit{size}(s) > 0$, we have $\mathit{size}(t') < \mathit{size}(t_i)$, which leads (by an inductive argument) to a contradiction eventually because $\mathit{size}(t)$ is finite.

Therefore, it cannot be the case that $t \notin L^\star$.

Hence, $L(\mathfrak{A}'') = L^\star$. $\square$

6.41 Remark The forest concatenation operation and the forest star operation are more general and more complex than the according tree operations. For that reason, it is more complex to show that the recognizable forest languages are closed under that operations than it is to show that the recognizable tree languages are closed under tree concatenation and tree concatenation closure.

6.42 This section can be concluded with the following interesting observation. Consider first a (non-ranked) alphabet A and the set A^* of all finite words over A. Then the set $Rec(A^*)$ of recognizable languages in A^*, together with the (binary) language concatenation operation (which is a straightforward extension of the word concatenation operation, see 2.14) and the language $\{\varepsilon\} \in Rec(A^*)$ as unit element, forms a monoid because the set $Rec(A^*)$ is closed under concatenation.
In a similar way, for a given ranked alphabet Σ, the $\mathbb{N} \times \mathbb{N}$-indexed set $Rec(\Sigma^{\circledast})$ (cf. 5.19), together with the concatenation operation (see Definition 6.27), the cartesian product (see Definition 6.9) as tupling function (i.e. for every $n, m \in \mathbb{N}$ and $L_1, \ldots, L_m \in Rec(\Sigma^{\circledast})[n,1]$, define $\langle\!\langle L_1, \ldots, L_m \rangle\!\rangle = L_1 \times \ldots \times L_m$) and the languages $\{\pi_i^n\} \in Rec(\Sigma^{\circledast})[n,1]$ (for every $n \in \mathbb{N}$ and $i = 1, \ldots, n$) as projections, forms a polyoid. The following proposition spells out the technical details.

6.43 Proposition *Let Σ be a ranked alphabet. Then, the tuple $\mathcal{R}(\Sigma^{\circledast}) = \langle Rec(\Sigma^{\circledast}), \cdot, \langle \ldots \rangle_{\mathcal{R}}, \Pi_{\mathcal{R}} \rangle$ is a polyoid, where*

- *the operation $\cdot : Rec(\Sigma^{\circledast}) \times Rec(\Sigma^{\circledast}) \dashrightarrow Rec(\Sigma^{\circledast})$ is the restriction of the concatenation operation to the recognizable languages,*
- *the tupling function*

$$\langle \ldots \rangle_{\mathcal{R}} : \left(\bigcup_{n,m \in \mathbb{N}} Rec(\Sigma^{\circledast})[n,1]^m \right) \to Rec(\Sigma^{\circledast}) : \langle L_1, \ldots, L_m \rangle \mapsto L_1 \times \ldots \times L_m$$

  maps an m-tuple of recognizable $(n,1)$-forest languages to their cartesian product, and
- *the $\mathbb{N}$-indexed set $\Pi_{\mathcal{R}}$ of projections is defined by*

$$\Pi_{n\mathcal{R}} = \{\ \{\pi_1^n\}, \ldots, \{\pi_n^n\}\ \} \text{ (for every } n \in \mathbb{N}\text{)},$$

  i.e. for every $n \geq 1$ and $i = 1, \ldots, n$, we have $\pi_{i\,\mathcal{R}}^n = \{\pi_i^n\}$.

Proof: We have to show that Conditions (i)-(iv) of Definition 4.5 are satisfied.

(i) Let $n, m \in \mathbb{N}$ and $L \in Rec(\Sigma^{\circledast})[n,m]$. Then

$$\langle\!\langle \pi_{1\,\mathcal{R}}^n, \ldots, \pi_{n\,\mathcal{R}}^n \rangle\!\rangle_{\mathcal{R}} \cdot L \;=\; (\{\pi_1^n\} \times \ldots \times \{\pi_n^n\}) \cdot L \;=\; \{\langle \pi_1^n, \ldots, \pi_n^n \rangle\} \cdot L \;=\; L.$$

Similarly, $L = L \cdot \langle\!\langle \pi_{1\,\mathcal{R}}^m, \ldots, \pi_{m\,\mathcal{R}}^m \rangle\!\rangle_{\mathcal{R}}$.

(ii) Proposition 6.33.

(iii) Immediate from Definitions 6.27 and 6.7, and Corollary 6.13.

(iv) Corollary 6.13.

□

7 Rational Forest Languages

In this section I will define the rational forest languages. After the groundwork of the previous section, this is straightforward and does not create any surprises. As in the case of word languages (cf. [Eil76], [Ber79], or [Per90]), there are two possible ways of introducing them: either by defining them using rational expressions, or by defining them to be the smallest family that contains the singletons and is closed under certain operations. In the case of forest languages, those operations are component-wise union, cartesian product, concatenation, and star. I will define them by the first way in Definition 7.3 and state the equivalence to the second in Proposition 7.6.

7.1 The formal definition of rational languages is split into two parts. First, I will define the sets of rational expressions, which are just sequences of symbols, i.e. purely syntactical. Then I will define the semantics of a rational expression, which, of course, will be a rational language.

7.2 Definition Let Σ be a ranked alphabet. Then the $\mathbb{N} \times \mathbb{N}$-indexed set $RAT(\Sigma^{\circledast})$ of *rational expressions* (over $\Sigma^{\circledast}$) is inductively defined as follows. Let n, m, m_1, m_2, k be natural numbers.

(i) If $m \leq 1$ and $L \subseteq \Sigma^{\circledast}[n, m]$ is finite and nonempty, then $L \in RAT(\Sigma^{\circledast})[n, m]$.

(ii) If $R_1, R_2 \in RAT(\Sigma^{\circledast})[n, 1]$, then $(R_1 \cup R_2) \in RAT(\Sigma^{\circledast})[n, 1]$.

(iii) If $R_1 \in RAT(\Sigma^{\circledast})[n, m_1]$ and $R_2 \in RAT(\Sigma^{\circledast})[n, m_2]$,
then $(R_1 \times R_2) \in RAT(\Sigma^{\circledast})[n, m_1 + m_2]$.

(iv) If $R_1 \in RAT(\Sigma^{\circledast})[n, m]$ and $R_2 \in RAT(\Sigma^{\circledast})[m, k]$, then $(R_1 \cdot R_2) \in RAT(\Sigma^{\circledast})[n, k]$.

(v) If $R \in RAT(\Sigma^{\circledast})[n, n]$, then $R^{\star} \in RAT(\Sigma^{\circledast})[n, n]$.

(vi) There are no other rational expression.

7.3 Definition Let Σ be a ranked alphabet. The *semantics of an rational expression* is a function

$$[\![-]\!] : RAT(\Sigma^{\circledast}) \to \bigcup_{n,m \in \mathbb{N}} \mathfrak{P}(\Sigma^{\circledast}[n, m]),$$

which is defined inductively as follows. Let $n, m, m_1, m_2, k \in \mathbb{N}$.

(i) If $m \leq 1$ and $L \subseteq \Sigma^{\circledast}[n, m]$ is finite and nonempty, then then $[\![L]\!] = L$.

(ii) If $R_1, R_2 \in RAT(\Sigma^{\circledast})[n, 1]$, then $[\![(R_1 \cup R_2)]\!] = [\![R_1]\!] \cup [\![R_2]\!]$.

(iii) If $R_1 \in RAT(\Sigma^{\circledast})[n, m_1]$ and $R_2 \in RAT(\Sigma^{\circledast})[n, m_2]$, then $[\![(R_1 \times R_2)]\!] = [\![R_1]\!] \times [\![R_2]\!]$.

(iv) If $R_1 \in RAT(\Sigma^{\circledast})[n, m]$ and $R_2 \in RAT(\Sigma^{\circledast})[m, k]$, then $[\![(R_1 \cdot R_2)]\!] = [\![R_1]\!] \cdot [\![R_2]\!]$.

(v) If $R \in RAT(\Sigma^{\circledast})[n,n]$, then $[\![R^\star]\!] = [\![R]\!]^\star$.

The $\mathbb{N}\times\mathbb{N}$-indexed set $Rat(\Sigma^{\circledast})$ of *rational languages* is defined to be the image of the function $[\![-]\!]$. In other words, for every $n, m \in \mathbb{N}$,

$$Rat(\Sigma^{\circledast})[n,m] = \{L \subseteq \Sigma^{\circledast}[n,m] \mid \exists R \in RAT(\Sigma^{\circledast})[n,m] \,.\, [\![R]\!] = L\}.$$

7.4 This simply says that a language is rational if and only if there exists a rational expression that has this language as its semantics. Hence, if one wants to show that a certain language is rational, it is usually the easiest way to show a rational expression for this language. Anyway, if one does not want to go the detour over rational expressions, there is another way of defining rational languages, which is stated in Proposition 7.6.

7.5 Lemma *Let Σ be a ranked alphabet and $n, m \in \mathbb{N}$. Further, let $L \subseteq \Sigma^{\circledast}[n,m]$. Then the following are equivalent.*

(i) $L \in Rat(\Sigma^{\circledast})[n,m]$.

(ii) There are $L_1, \dots, L_m \in Rat(\Sigma^{\circledast})[n,1]$, such that $L = L_1 \times \ldots \times L_m$.

Proof: First, assume that there are $L_1, \dots, L_m \in Rat(\Sigma^{\circledast})[n,1]$. Then it follows immediately from Definitions 7.2 and 7.3 that $L = L_1 \times \ldots \times L_m \in Rat(\Sigma^{\circledast})[n,m]$.
For the other direction, let $L \in Rat(\Sigma^{\circledast})[n,m]$. Then there is a rational expression $R \in RAT(\Sigma^{\circledast})[n,m]$ with $[\![R]\!] = L$. By induction on the structure of R, I will now show that $[\![R]\!] = L_1 \times \ldots \times L_m$ with $L_1, \dots, L_m \in Rat(\Sigma^{\circledast})[n,1]$.

(i) The case $m \le 1$ and $R \subseteq \Sigma^{\circledast}[n,m]$ finite and nonempty is trivial.

(ii) The case $R = (R_1 \cup R_2)$ with $R_1, R_2 \in RAT(\Sigma^{\circledast})[n,1]$ is also trivial.

(iii) Let $R = (R_1 \times R_2)$ with $R_1 \in RAT(\Sigma^{\circledast})[n,m_1]$ and $R_2 \in RAT(\Sigma^{\circledast})[n,m_2]$ for some $n, m_1, m_2 \in \mathbb{N}$. By induction hypothesis, $[\![R_1]\!] = L_{11} \times \ldots \times L_{1m_1}$ and $[\![R_2]\!] = L_{21} \times \ldots \times L_{2m_2}$. By Definition 7.3, $[\![R]\!] = [\![(R_1 \times R_2)]\!] = [\![R_1]\!] \times [\![R_2]\!] = L_{11} \times \ldots \times L_{1m_1} \times L_{21} \times \ldots \times L_{2m_2}$.

(iv) Let $R = (R_1 \cdot R_2)$ with $R_1 \in RAT(\Sigma^{\circledast})[n,k]$ and $R_2 \in RAT(\Sigma^{\circledast})[k,m]$ for some $n, m, k \in \mathbb{N}$. By induction hypothesis, $[\![R_2]\!] = L_{21} \times \ldots \times L_{2m}$. By Definition 7.3 and Lemma 6.29, $[\![R]\!] = [\![(R_1 \cdot R_2)]\!] = [\![R_1]\!] \cdot (L_{21} \times \ldots \times L_{2m}) = [\![R_1]\!] \cdot L_{21} \times \ldots \times [\![R_1]\!] \cdot L_{2m}$.

(v) Let $R = R_1^\star$ with $R_1 \in RAT(\Sigma^{\circledast})[n,n]$ for some $n \in \mathbb{N}$. By Lemma 6.39, $L = [\![R]\!] = [\![R_1]\!]^\star = L_1 \times \ldots \times L_n$ for some $L_1, \dots, L_n \subseteq \Sigma^{\circledast}[n,1]$. Further, $L = [\![R]\!] = [\![R]\!];\pi_1^n \times \ldots \times [\![R]\!];\pi_n^n = [\![R]\!] \cdot \pi_1^n \times \ldots \times [\![R]\!] \cdot \pi_n^n = [\![R]\!] \cdot [\![\{\pi_1^n\}]\!] \times \ldots \times [\![R]\!] \cdot [\![\{\pi_n^n\}]\!] = [\![R \cdot \{\pi_1^n\}]\!] \times \ldots \times [\![R \cdot \{\pi_n^n\}]\!]$. Hence, for every $i = 1, \dots, n$, we have $L_i = [\![R \cdot \{\pi_i^n\}]\!]$, where $R \cdot \{\pi_i^n\}$ is a rational expression. Therefore for every $i = 1, \dots, n$, the language L_i is rational.

□

7.6 Proposition *Let Σ be a ranked alphabet. Then the set $Rat(\Sigma^{\circledast})$ is the smallest $\mathbb{N}\times\mathbb{N}$-indexed set $\mathcal{S}$ such that*

(i) for every $n, m \in \mathbb{N}$ and $t \in \Sigma^{\circledast}[n,m]$, the singleton $\{t\} \in \mathcal{S}[n,m]$,

(ii) for every $n, m \in \mathbb{N}$, if $L_1, L_2 \in \mathcal{S}[n,m]$, then $L_1 \uplus L_2 \in \mathcal{S}[n,m]$,

(iii) for every $n, m_1, m_2 \in \mathbb{N}$, if $L_1 \in \mathcal{S}[n, m_1]$ and $L_2 \in \mathcal{S}[n, m_2]$, then $L_1 \times L_2 \in \mathcal{S}[n, m_1+m_2]$,

(iv) for every $n, k, m \in \mathbb{N}$, if $L_1 \in \mathcal{S}[n, m]$ and $L_2 \in \mathcal{S}[m, k]$, then $L_1 \cdot L_2 \in \mathcal{S}[n, k]$, and

(v) for every $n \in \mathbb{N}$, if $L \in \mathcal{S}[n, n]$, then $L^\star \in \mathcal{S}[n, n]$.

Proof: Let me first show that for every $n, m \in \mathbb{N}$, we have $\mathcal{S}[n, m] \subseteq Rat(\Sigma^\circledast)[n, m]$, or in other words, that the $\mathbb{N} \times \mathbb{N}$-indexed set $Rat(\Sigma^\circledast)$ fulfills Conditions (i)-(iv).

(i) Let $m \leq 1$. Then the singletons are in $Rat(\Sigma^\circledast)[n, m]$ because they are finite. For $m \geq 2$, let $t \in \Sigma^\circledast[n, m]$. Then $t = \langle t_1, \ldots, t_m \rangle$ for $t_1, \ldots, t_m \in \Sigma^\circledast[n, 1]$. Hence $\{t\} = \{t_1\} \times \ldots \times \{t_m\} \in Rat(\Sigma^\circledast)[n, m]$ because $\{t_1\}, \ldots, \{t_m\} \in Rat(\Sigma^\circledast)[n, 1]$.

(ii) Let $L_1, L_2 \in Rat(\Sigma^\circledast)[n, m]$. Then by Lemma 7.5, $L_1 = L_{11} \times \ldots \times L_{1m}$ and $L_2 = L_{21} \times \ldots \times L_{2m}$ where $L_{11}, \ldots, L_{1m}, L_{21}, \ldots, L_{2m} \in Rat(\Sigma^\circledast)[n, 1]$. Furthermore, we have $L_1 \uplus L_2 = (L_{11} \cup L_{21}) \times \ldots \times (L_{1m} \cup L_{2m})$, which is obviously rational.

(iii) Immediate from Definitions 7.2 and 7.3.

(iv) Immediate from Definitions 7.2 and 7.3.

(v) Immediate from Definitions 7.2 and 7.3.

For the other direction, let $L \in Rat(\Sigma^\circledast)[n, m]$ for some $n, m \in \mathbb{N}$. Then it follows from an easy induction on the structure of the rational expression $R \in RAT(\Sigma^\circledast)[n, m]$ with $[\![R]\!] = L$ that $L \in \mathcal{S}[n, m]$, because the finite languages (for $m \leq 1$) are obtained from the singletons by applying the $\uplus$ operation, and $\cup$ is the restriction of $\uplus$ to $m = 1$. $\square$

7.7 I excluded the empty set from the class $Rat(\Sigma^\circledast)$, because otherwise we would not get an $\mathbb{N} \times \mathbb{N}$-indexed set. Observe that for every ranked alphabet Σ, the $\mathbb{N} \times \mathbb{N}$-indexed set $Rat(\Sigma^\circledast)$ forms a polyoid. I will not give a proof of this statement here because it is a straightforward simplification of the proof of Proposition 6.43.

7.8 The reason for excluding the empty set was purely technical. In fact, from a pragmatic point of view, the empty set should be considered to be a rational language. For that reason, let me define for every ranked alphabet Σ and natural numbers n and m the set

$$Rat(\Sigma^\circledast, n, m) \;=\; Rat(\Sigma^\circledast)[n, m] \cup \{\emptyset\}.$$

Note that by adding the empty set, the rational languages do not lose their closure properties. In particular, Lemma 7.5 does also hold for every $L \in Rat(\Sigma^\circledast, n, m)$.
Now, it is time to state the main theorem of this thesis.

7.9 Theorem *If Σ is a ranked alphabet, and n and m are natural numbers, then*

$$Rat(\Sigma^\circledast, n, m) = Rec(\Sigma^\circledast, n, m).$$

Proof: The proof of this theorem splits into two parts because we have to show both directions of the equality. For the first direction, all the work has already been done in the previous section. Let $L \subseteq \Sigma^\circledast[n, m]$ and assume that $L \in Rat(\Sigma^\circledast, n, m)$. By an induction argument and the use of Propositions 6.1, 6.4, 6.12, 6.34, and 6.40, it follows that $L \in Rec(\Sigma^\circledast, n, m)$.

For the other direction, let $L \in Rec(\Sigma^\circledast, n, m)$. By Proposition 6.12, $L = L_1 \times \ldots \times L_m$ with $L_1, \ldots, L_m \in Rec(\Sigma^\circledast, n, 1)$. Therefore, in order to show that $L \in Rat(\Sigma^\circledast, n, m)$ it suffices to show that $L_1, \ldots, L_m \in Rat(\Sigma^{\circledast}, n, 1)$ (by Lemma 7.5). Hence, the proof can be restricted to the case $m = 1$. So, let $\mathfrak{A} = \langle Q, \Sigma, I, F, E \rangle$ be an $(n,1)$-fsfa (i.e. $I = I_1 \times \ldots \times I_n$ with $I_1, \ldots, I_n \subseteq Q$ and $F \subseteq Q$). I will now show, that $L(\mathfrak{A}) \in Rat(\Sigma^{\circledast}, n, 1)$. Without loss of generality, assume that $Q = \{1, \ldots, z\}$ with $z \in \mathbb{N}$. Now, define for every $\zeta \in \{0, \ldots, z\}$ and $q \in \{1, \ldots, z\}$, the language $L_{\zeta,q} \subseteq \Sigma^{\circledast}[n,1]$ as follows.

$$\begin{aligned} t \in L_{\zeta,q} \quad \Longleftrightarrow \quad & (\ \exists i \in \{1, \ldots, n\}\ .\ t = \pi_i^n \wedge q \in I_i\ )\ \vee \\ & (\ \exists k \in \mathbb{N}\ .\ \exists q_1, \ldots, q_k \leq \zeta\ .\ \exists t_1 \in L_{\zeta,q_1}\ \ldots\ \exists t_k \in L_{\zeta,q_k}\ .\ \exists \sigma \in \Sigma_k\ . \\ & \quad t = \langle t_1, \ldots, t_k \rangle \sigma \wedge (\langle q_1, \ldots, q_k \rangle, \sigma, q) \in E_k\ ) \end{aligned}$$

It is easy to see that for every $t \in \Sigma^{\circledast}[n,1]$,

$$t \in L_{z,q} \iff (\langle I_1, \ldots, I_n \rangle, t, q) \in E^{\mathfrak{A}}_{n,1}.$$

Hence,

$$L(\mathfrak{A}) \quad = \quad \bigcup_{q \in F} L_{z,q}. \tag{14}$$

Therefore it suffices to show that $L_{z,q}$ is rational for every $q \in F$. However, I will now show that for every $\zeta \in \{0, \ldots, z\}$ and $q \in \{1, \ldots, z\}$, the set $L_{\zeta,q} \in Rat(\Sigma^{\circledast}, n, 1)$. This will be done by induction on ζ.

- Induction Base ($\zeta = 0$): The set

  $$L_{0,q} \quad = \quad \{\pi_i^n \mid q \in I_i\} \cup \{\langle\rangle_n \sigma \mid \sigma \in \Sigma_0\ \wedge (\langle\rangle, \sigma, q) \in E_0\} \tag{15}$$

  is rational because it is finite.

- Induction Step: Assume $L_{\zeta,q}$ is rational for every $q \in \{1, \ldots, z\}$. In order to show that $L_{\zeta+1,q}$ is also rational, we need to define another auxiliary family of sets. For every $\xi \in \{0, \ldots, z\}$ and $p \in \{1, \ldots, z\}$ define the language $L^E_{\xi,p} \subseteq \Sigma^{\circledast}[\xi, 1]$ as follows.

  $$\begin{aligned} L^E_{\xi,p} \quad &= \quad \bigcup_{k \geq 0} L^{E_k}_{\xi,p}, \text{ where for every } k \geq 0 \\ L^{E_k}_{\xi,p} \quad &= \quad \{\langle \pi^{\xi}_{q_1}, \ldots, \pi^{\xi}_{q_k} \rangle \sigma \mid \sigma \in \Sigma_k \wedge (\langle q_1, \ldots, q_k \rangle, \sigma, p) \in E_k \wedge q_1, \ldots, q_k \leq \xi\} \end{aligned}$$

  Then $L^E_{\xi,p}$ is certainly rational because it is finite. Now the induction argument is the following equation.

  $$L_{\zeta+1,q} \quad = \quad L_{\zeta,q} \cup (L_{\zeta,1} \times \ldots \times L_{\zeta,\zeta+1}) \cdot (L^E_{\zeta+1,1} \times \ldots \times L^E_{\zeta+1,\zeta+1})^{\star} \cdot L^E_{\zeta+1,q} \tag{16}$$

  This implies that $L_{\zeta+1,q} \in Rat(\Sigma^{\circledast}, n, 1)$ by the definition of rational languages. Hence, in order to complete the proof of the theorem, I have to show that (16) holds.

  - For the first inclusion, let $t \in L_{\zeta+1,q}$. By structural induction on t, I will show that

    $$t \quad \in \quad L_{\zeta,q} \cup (L_{\zeta,1} \times \ldots \times L_{\zeta,\zeta+1}) \cdot (L^E_{\zeta+1,1} \times \ldots \times L^E_{\zeta+1,\zeta+1})^{\star} \cdot L^E_{\zeta+1,q}. \tag{17}$$

    * $t = \pi_i^n$, for some $i \in \{1, \ldots, n\}$. Then $t \in L_{\zeta,q}$.
    * $t = \langle t_1, \ldots, t_k \rangle \sigma$, for some $k \geq 0$, $\sigma \in \Sigma_k$ and $t_1, \ldots, t_k \in \Sigma^{\circledast}[n,1]$. Then there are two cases, either $t \in L_{\zeta,q}$ or $t \notin L_{\zeta,q}$. In the first case, (17) trivially holds. So, consider the second case. Since $t = \langle t_1, \ldots, t_k \rangle \sigma \in L_{\zeta+1,q}$, there are $q_1, \ldots, q_k \leq \zeta + 1$, such that $(\langle q_1, \ldots, q_k \rangle, \sigma, q) \in E_k$ and $t_i \in L_{\zeta+1,q_i}$ for every $i = 1, \ldots, k$. Hence

      $$\langle \pi^{\zeta+1}_{q_1}, \ldots, \pi^{\zeta+1}_{q_k} \rangle \sigma \quad \in \quad L^E_{\zeta+1,q} \tag{18}$$

      and $t_i \in L_{\zeta,q_i} \cup (L_{\zeta,1} \times \ldots \times L_{\zeta,\zeta+1}) \cdot (L^E_{\zeta+1,1} \times \ldots \times L^E_{\zeta+1,\zeta+1})^{\star} \cdot L^E_{\zeta+1,q_i}$ for every $i = 1, \ldots, k$ (by induction hypothesis).

· If $t_i \in L_{\zeta,q_i}$, then $t_i \in (L_{\zeta,1} \times \ldots \times L_{\zeta,\zeta+1}); \pi_{q_i}^{\zeta+1}$, and therefore also

$$t_i \in (L_{\zeta,1} \times \ldots \times L_{\zeta,\zeta+1}) \cdot ((L^E_{\zeta+1,1} \times \ldots \times L^E_{\zeta+1,\zeta+1})^\star; \pi_{q_i}^{\zeta+1}).$$

· If $t_i \in (L_{\zeta,1} \times \ldots \times L_{\zeta,\zeta+1}) \cdot (L^E_{\zeta+1,1} \times \ldots \times L^E_{\zeta+1,\zeta+1})^\star \cdot L^E_{\zeta+1,q_i}$, then there is an $s \in (L^E_{\zeta+1,1} \times \ldots \times L^E_{\zeta+1,\zeta+1})^\star \cdot L^E_{\zeta+1,q_i}$ with $t_i \in (L_{\zeta,1} \times \ldots \times L_{\zeta,\zeta+1}) \cdot s$. Since $q_i \leq \zeta + 1$, it follows that $s \in (L^E_{\zeta+1,1} \times \ldots \times L^E_{\zeta+1,\zeta+1})^\star; \pi_{q_i}^{\zeta+1}$. Therefore,

$$t_i \in (L_{\zeta,1} \times \ldots \times L_{\zeta,\zeta+1}) \cdot ((L^E_{\zeta+1,1} \times \ldots \times L^E_{\zeta+1,\zeta+1})^\star; \pi_{q_i}^{\zeta+1}).$$

Hence, for every $i = 1, \ldots, k$ it holds that

$$t_i \quad \in \quad ((L_{\zeta,1} \times \ldots \times L_{\zeta,\zeta+1}) \cdot (L^E_{\zeta+1,1} \times \ldots \times L^E_{\zeta+1,\zeta+1})^\star); \pi_{q_i}^{\zeta+1}. \tag{19}$$

From (19) and (18) it immediately follows that

$$t = \langle t_1, \ldots, t_k \rangle \sigma \in (L_{\zeta,1} \times \ldots \times L_{\zeta,\zeta+1}) \cdot (L^E_{\zeta+1,1} \times \ldots \times L^E_{\zeta+1,\zeta+1})^\star \cdot L^E_{\zeta+1,q}.$$

Therefore, $L_{\zeta+1,q} \subseteq L_{\zeta,q} \cup (L_{\zeta,1} \times \ldots \times L_{\zeta,\zeta+1}) \cdot (L^E_{\zeta+1,1} \times \ldots \times L^E_{\zeta+1,\zeta+1})^\star \cdot L^E_{\zeta+1,q}$

– For the other inclusion, let $t \in L_{\zeta,q} \cup (L_{\zeta,1} \times \ldots \times L_{\zeta,\zeta+1}) \cdot (L^E_{\zeta+1,1} \times \ldots \times L^E_{\zeta+1,\zeta+1})^\star \cdot L^E_{\zeta+1,q}$. If $t \in L_{\zeta,q}$, then trivially $t \in L_{\zeta+1,q}$. If $t \in (L_{\zeta,1} \times \ldots \times L_{\zeta,\zeta+1}) \cdot (L^E_{\zeta+1,1} \times \ldots \times L^E_{\zeta+1,\zeta+1})^\star \cdot L^E_{\zeta+1,q}$, then it follows from the definition of $L_{\zeta+1,q}$ that $t \in L_{\zeta+1,q}$. This is easy to see: Assume (by way of contradiction) that $t \notin L_{\zeta+1,q}$. Then a state $q' > \zeta + 1$ must occur eventually when the automaton $\mathfrak{A}$ is processing the forest t. But whenever the automaton is processing a forest $t' \in (L_{\zeta,1} \times \ldots \times L_{\zeta,\zeta+1}) \cdot (L^E_{\zeta+1,1} \times \ldots \times L^E_{\zeta+1,\zeta+1})^\star \cdot L^E_{\zeta+1,q}$, only states $q'' \leq \zeta + 1$ are visited. Therefore $t \notin (L_{\zeta,1} \times \ldots \times L_{\zeta,\zeta+1}) \cdot (L^E_{\zeta+1,1} \times \ldots \times L^E_{\zeta+1,\zeta+1})^\star \cdot L^E_{\zeta+1,q}$, which is a contradiction to the assumption.
Therefore, $L_{\zeta+1,q} \supseteq L_{\zeta,q} \cup (L_{\zeta,1} \times \ldots \times L_{\zeta,\zeta+1}) \cdot (L^E_{\zeta+1,1} \times \ldots \times L^E_{\zeta+1,\zeta+1})^\star \cdot L^E_{\zeta+1,q}$.

This completes the proof of (16).

Hence, for every $\zeta \in \{0, \ldots, z\}$ and $q \in \{1, \ldots, z\}$, we have $L_{\zeta,q} \in Rat(\Sigma^\circledast, n, 1)$. Therefore, by (14), we have $L(\mathfrak{A}) \in Rat(\Sigma^\circledast, n, 1)$. □

7.10 Example Consider again the automaton $\mathfrak{A} = \langle Q, \Sigma, I, F, E \rangle$ from Example 5.5. According to the proof of Theorem 7.9, rename the states e and o to 1 and 2, respectively.

$$\begin{aligned}
Q' &= \{1, 2\} \\
I' &= \{\langle 1, 1, 1 \rangle\} \\
F' &= \{1\} \\
E'_0 &= \{(\langle\rangle, \alpha, 2)\} \\
E'_2 &= \{(\langle 1, 1 \rangle, \sigma, 2), (\langle 1, 2 \rangle, \sigma, 1), (\langle 2, 2 \rangle, \sigma, 2), (\langle 2, 1 \rangle, \sigma, 1)\}
\end{aligned}$$

The languages $L^E_{\xi,p}$ are now

$$\begin{aligned}
L^E_{0,1} &= \emptyset \\
L^E_{0,2} &= \{\langle\rangle_0 \alpha\} \\
L^E_{1,1} &= \emptyset \\
L^E_{1,2} &= \{\langle\rangle_1 \alpha, \langle \pi^1_1, \pi^1_1 \rangle \sigma\} \\
L^E_{2,1} &= \{\langle \pi^2_1, \pi^2_2 \rangle \sigma, \langle \pi^2_2, \pi^2_1 \rangle \sigma\} \\
L^E_{2,2} &= \{\langle \pi^2_1, \pi^2_1 \rangle \sigma, \langle \pi^2_2, \pi^2_2 \rangle \sigma, \langle\rangle_2 \alpha\}
\end{aligned}$$

According to (15) and (16), we can compute the languages $L_{\zeta,q}$.

$$\begin{array}{rcl} L_{0,1} & = & \{\pi_1^3, \pi_2^3, \pi_3^3\} \\ L_{0,2} & = & \{\langle\rangle_3\alpha\} \\ L_{1,1} & = & L_{0,1} \cup L_{0,1} \cdot {L_{1,1}^E}^\star \cdot L_{1,1}^E \\ & = & \{\pi_1^3, \pi_2^3, \pi_3^3\} \\ L_{1,2} & = & L_{0,2} \cup L_{0,1} \cdot {L_{1,1}^E}^\star \cdot L_{1,2}^E \\ & = & \{\langle\rangle_3\alpha, \langle\pi_1^3, \pi_1^3\rangle\sigma, \langle\pi_1^3, \pi_2^3\rangle\sigma, \langle\pi_1^3, \pi_3^3\rangle\sigma, \ldots, \langle\pi_3^3, \pi_3^3\rangle\sigma\} \\ L_{2,1} & = & L_{1,1} \cup (L_{1,1} \times L_{1,2}) \cdot (L_{2,1}^E \times L_{2,2}^E)^\star \cdot L_{2,1}^E \\ & = & \{\pi_1^3, \pi_2^3, \pi_3^3\} \cup \\ & & (\{\pi_1^3, \pi_2^3, \pi_3^3\} \times \{\langle\rangle_3\alpha, \langle\pi_1^3, \pi_1^3\rangle\sigma, \langle\pi_1^3, \pi_2^3\rangle\sigma, \langle\pi_1^3, \pi_3^3\rangle\sigma, \ldots, \langle\pi_3^3, \pi_3^3\rangle\sigma\}) \cdot \\ & & (\{\langle\pi_1^2, \pi_2^2\rangle\sigma, \langle\pi_2^2, \pi_1^2\rangle\sigma\} \times \{\langle\pi_1^2, \pi_1^2\rangle\sigma, \langle\pi_2^2, \pi_2^2\rangle\sigma, \langle\rangle_2\alpha\})^\star \cdot \\ & & \{\langle\pi_1^2, \pi_2^2\rangle\sigma, \langle\pi_2^2, \pi_1^2\rangle\sigma\} \end{array}$$

Then, $L(\mathfrak{A}) = L_{2,1}$.

7.11 Corollary *Let Σ be a ranked alphabet. Then*

$$Rat(\Sigma^{\circledast}) \;=\; Rec(\Sigma^{\circledast}).$$

Proof: It follows immediately from Theorem 7.9 that for every $n, m \in \mathbb{N}$,

$$Rat(\Sigma^{\circledast})[n, m] = Rec(\Sigma^{\circledast})[n, m].$$

□

7.12 A natural question that arises now is: How is this result related to the theorem by Kleene [Kle56] and to the theorem by Thatcher and Wright [TW68]? The answer is: It depends on what is allowed under the star.

In order to explain this, let me introduce a subclass of the rational forest languages. Define for every ranked alphabet Σ and natural number n_0, the $\mathbb{N} \times \mathbb{N}$-indexed set n_0-$Rat(\Sigma^{\circledast})$ similar to $Rat(\Sigma^{\circledast})$ with the only difference that the star can only be applied to rational languages $L \subseteq \Sigma^{\circledast}[n, n]$ with $n \leq n_0$.

In order to restate Kleene's theorem for word languages, consider a ranked alphabet Σ with $\Sigma = \Sigma_1$ (i.e. all symbols $\sigma \in \Sigma$ have rank 1). Then Σ^* and $\Sigma^{\circledast}[1, 1]$ are isomorphic and Kleene's star operation in Σ^* corresponds to the forest star operation in $\Sigma^{\circledast}[1, 1]$. With this in mind, Kleene's theorem can spelled out in the formalism of this thesis.

7.13 Theorem *(Kleene, 1956) For every ranked alphabet Σ with $\Sigma = \Sigma_1$,*

$$1\text{-}Rat(\Sigma^{\circledast}, 1, 1) = Rec(\Sigma^{\circledast}, 1, 1).$$

This means that Theorem 7.9 is not a generalization of Kleene's theorem. But this should not be considered to be a problem because the result by Thatcher and Wright is neither a generalization of Kleene's theorem. A true generalization Kleene's theorem would be for instance the following statement, which I could neither prove nor disprove.

7.14 Conjecture *For every natural number n and ranked alphabet Σ with $\Sigma = \Sigma_n$,*

$$n\text{-}Rat(\Sigma^{\circledast}, n, n) = Rec(\Sigma^{\circledast}, n, n).$$

7.15 However, Theorem 7.9 can be seen as a generalization of the result by Thatcher and Wright. To see this let me also restate that result in the formalism of this thesis. This makes it necessary to define another subclass of the rational forest languages, namely the *s-rational* forest languages (the "s" stands for "single"). For every ranked alphabet Σ define the $\mathbb{N} \times \mathbb{N}$-indexed set $s\text{-}Rat(\Sigma^{\circledast})$ similar to $Rat(\Sigma^{\circledast})$ with the difference that the star operation can only be applied to rational languages $L \subseteq \Sigma^{\circledast}[n, n]$ where $n \in \mathbb{N}$ and $L = L_1 \times \ldots \times L_n$ such that there is exactly one $i \in \{1, \ldots, n\}$ with $L_i \neq \{\pi_i^n\}$. Then the star operation is reduced to the tree concatenation closure as it is defined in [TW68]. Now, it follows from [TW68] that

$$Rec(\Sigma^{\circledast}, 0, 1) \subseteq s\text{-}Rat(\Sigma^{\circledast}, 0, 1),$$

because the tree concatenation operation (defined in [TW68]) is subsumed by the forest concatenation operation (Definition 6.27). Together with the results from Section 6, we get

7.16 Theorem *(Thatcher and Wright, 1968) For every ranked alphabet Σ,*

$$s\text{-}Rat(\Sigma^{\circledast}, 0, 1) = Rec(\Sigma^{\circledast}, 0, 1).$$

7.17 Example There are some technical subtleties connected with this result. Instead of giving a complicated proof, let me use the example from the introduction to explain the problems. Consider the $(0, 1)$-fsfa $\mathfrak{A} = \langle Q, \Sigma, I, F, E\rangle$ with $E = \{E_k \mid k \geq 0\}$ such that

$$\begin{aligned}
Q &= \{q, f\}, \\
\Sigma &= \Sigma_0 \cup \Sigma_1 \cup \Sigma_2, \quad \text{where } \Sigma_0 = \{\alpha\},\ \Sigma_1 = \{\gamma\}, \text{ and } \Sigma_2 = \{\sigma\}, \\
I &= \{\langle\rangle\}, \\
F &= \{f\}, \\
E_0 &= \{(\langle\rangle, \alpha, p), (\langle\rangle, \alpha, f)\}, \\
E_1 &= \{(\langle p\rangle, \gamma, p)\}, \text{ and} \\
E_2 &= \{(\langle p, f\rangle, \sigma, f)\}.
\end{aligned}$$

The $(0, 1)$-forest language $L = L(\mathfrak{A})$ recognized by this automaton is shown in Figure 6. The rational expression (2) for tree languages can literally be translated into

$$L' = \{\langle \pi_1^2, \langle\rangle_2\alpha\rangle\} \cup \{\langle \pi_1^2, \langle\rangle_2\alpha\rangle\} \cdot \left(\{\langle\langle\rangle_2\alpha, \pi_2^2\rangle\} \cdot \{\langle \pi_1^2\gamma, \pi_2^2\rangle\}^{\star} \cdot \{\langle \pi_1^2, \langle \pi_1^2, \pi_2^2\rangle\sigma\rangle\}\right)^{\star}, \quad (20)$$

which is shown in Figure 11, where the braces are omitted for better readability.
There are two things to observe.

- The order of the operands of the concatenation operation is reversed. The reason lies in the way how trees and forests are written down. In the case of trees, the second operand is plugged into the leaves of the first operand, whereas in the case of forests, the first operand is plugged into the second. In our example, the first concatenation simulates the operation $\cdot_p$ in (2) and the other two concatenations simulate $\cdot_f$. Similarly, the two applications of the star simulate the operations ^{*f} and ^{*p}, respectively.

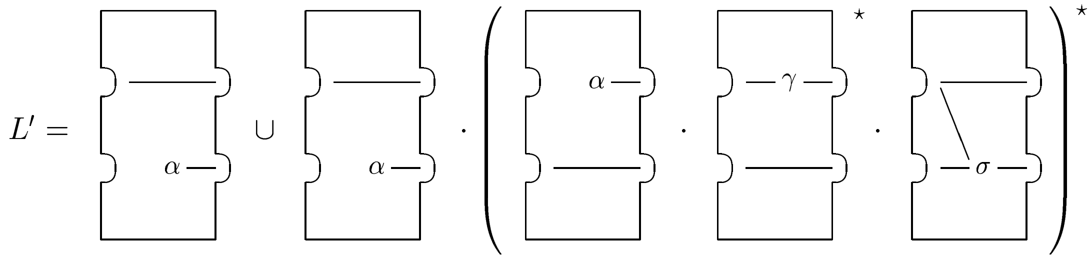

Figure 11: The rational expression (2) in forest notation.

- The language L' is a $(2,2)$-forest language. In order to obtain the $(0,1)$-forest language L, we have to concatenate L' with some $(0,2)$-forest language L'' on the left and with π_2^2 on the right:

$$L \quad = \quad \{\langle\langle\rangle_0\alpha, \langle\rangle_0\alpha\rangle\} \cdot L' \cdot \pi_2^2. \tag{21}$$

7.18 By some technical modification of the proof in [TW68], Theorem 7.16 can be extended to $s\text{-}Rat(\Sigma^{\circledast}, n, 1) = Rec(\Sigma^{\circledast}, n, 1)$ for every $n \in \mathbb{N}$. By Proposition 6.12, it follows that for every $n, m \in \mathbb{N}$,

$$s\text{-}Rat(\Sigma^{\circledast}, n, m) = Rec(\Sigma^{\circledast}, n, m).$$

This raises the question, why the proof of Theorem 7.9 is interesting. There are two points. First, the detour of (21) is avoided and the proof of Theorem 7.9 is simpler than the proof in [TW68]. To see the second, let me introduce another subclass of the rational forest languages, namely the *f-rational* forest languages (the "f" stands for "finite"). Define for every ranked alphabet Σ the $\mathbb{N}\times\mathbb{N}$-indexed set $f\text{-}Rat(\Sigma^{\circledast})$ similar to $Rat(\Sigma^{\circledast})$ with the difference that the star operation can only be applied to rational languages that are finite. This seems to be a rather strong restriction. However, if we look again at the proof of Theorem 7.9, we can see that in the induction step in (16), the star is applied only to cartesian products of languages $L^E_{\xi,p}$ which are finite. Hence, for every $n \in \mathbb{N}$,

$$Rec(\Sigma^{\circledast}, n, 1) \subseteq f\text{-}Rat(\Sigma^{\circledast}, n, 1).$$

7.19 Example Let me use the same language as in Example 7.17 to illustrate this correlation. Following the construction in the proof of Theorem 7.9, we obtain

$$\begin{aligned}
L \quad &= \quad \{\langle\rangle_0\alpha\} \;\cup \\
&\qquad \big(\{\langle\rangle_0\alpha\} \cdot \{\pi_1^1\gamma\}^\star \times \{\langle\rangle_0\alpha\}\big) \cdot \big(\{\langle\rangle_2\alpha, \pi_1^2\gamma\} \times \{\langle\rangle_2\alpha, \langle\pi_1^2, \pi_2^2\rangle\sigma\}\big)^\star \cdot \{\langle\rangle_2\alpha, \langle\pi_1^2, \pi_2^2\rangle\sigma\} \\
&= \quad \{\langle\rangle_0\alpha\} \;\cup \\
&\qquad \big(\{\langle\rangle_0\alpha\} \cdot \{\pi_1^1\gamma\}^\star \times \{\langle\rangle_0\alpha\}\big) \cdot \\
&\qquad \{\langle\langle\rangle_2\alpha, \langle\rangle_2\alpha\rangle, \langle\langle\rangle_2\alpha, \langle\pi_1^2, \pi_2^2\rangle\sigma\rangle, \langle\pi_1^2\gamma, \langle\rangle_2\alpha\rangle, \langle\pi_1^2\gamma, \langle\pi_1^2, \pi_2^2\rangle\sigma\rangle\}^\star \cdot \{\langle\pi_1^2, \pi_2^2\rangle\sigma\},
\end{aligned}$$

which is shown in Figure 12. In this example, the rational expression can be further simplified to the expression shown in Figure 7. But in general, such simplifications are not possible (see e.g. Example 7.10).

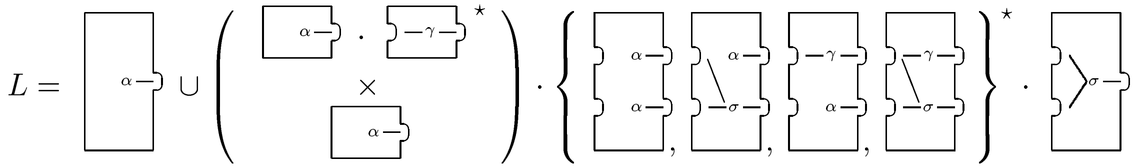

Figure 12: The rational expression for the language L shown in Figure 6 — obtained from the construction of the proof of Theorem 7.9.

7.20 Putting all the results together, we obtain the interesting fact that the s-rational forest languages and the f-rational forest languages are the same and that they do not form a proper subclass of the rational forest languages.

7.21 Theorem *For every ranked alphabet Σ and natural numbers n, m,*

$$Rat(\Sigma^{\circledast}, n, m) = s\text{-}Rat(\Sigma^{\circledast}, n, m) = f\text{-}Rat(\Sigma^{\circledast}, n, m) = Rec(\Sigma^{\circledast}, n, m).$$

7.22 Corollary *For every ranked alphabet Σ,*

$$Rat(\Sigma^{\circledast}) = s\text{-}Rat(\Sigma^{\circledast}) = f\text{-}Rat(\Sigma^{\circledast}) = Rec(\Sigma^{\circledast}).$$

7.23 Remark In [Eil76] it has been shown, how the notion of rational language can be extended from the free monoid to any monoid. In a similar way, it is possible to define rational languages in any polyoid by extending Definitions 7.2 and 7.3. There might occur a problem in 7.3.(iv) because I defined the concatenation operation for languages only in the free polyoid. But it should be possible to find a similar operation in every polyoid. However, further investigation into this problem would have exceeded the scope of this thesis.

8 Semi-Recognizable Forest Languages

In the previous three sections, I considered only languages $L \subseteq \Sigma^{\circledast}[n, m]$ that have the form $L = L_1 \times \ldots \times L_m$. Then L is recognizable if and only if $L_1, \ldots, L_m \subseteq \Sigma^{\circledast}[n, 1]$ are recognizable. This raises the question whether it is possible to find a notion of recognizability for languages which do not have this special form. This section gives a positive answer to that question. The idea is essentially the same as in Section 5. The only difference lies in the definition of the final state tuples of an automaton. In Section 5, the final state tuples had to be of the form $F = F_1 \times \ldots \times F_m$ with $F_1, \ldots, F_m \subseteq Q$. Now, the set of final state tuples can be any subset $F \subseteq Q^m$. The class of forest languages which are obtained that way will be called semi-recognizable. Compared to the recognizable languages, they loose some closure properties, but gain some other closure properties. In particular, the very important power set construction in the proof of Proposition 5.17 is also possible for semi-recognizable languages (see Proposition 8.8).

8.1 Definition Let $n, m \in \mathbb{N}$. A *(nondeterministic) semi finite-state* (n, m) *forest automaton* (*semi-*(n, m)*-fsfa*) is a tuple $\mathfrak{A} = \langle Q, \Sigma, I, F, E \rangle$, where Q is a finite set of *states*, Σ is a ranked alphabet, $I = I_1 \times \ldots \times I_n$ (where $I_1, \ldots, I_n \subseteq Q$) is the set of *initial state tuples*, $F \subseteq Q^m$ is the set of *final state tuples*, and

$$E = \{E_k \mid k \geq 0\}, \quad \text{where} \quad E_k \subseteq Q^k \times \Sigma_k \times Q \quad (\text{for every } k \geq 0)$$

is the set of *state transition relations*.

8.2 The definition of the relations $E^{\mathfrak{A}}_{k,l} \subseteq (\mathfrak{P}(Q))^k \times \Sigma^{\circledast}[k,l] \times Q^l$ (for every $k, l \in \mathbb{N}$) is carried out in exactly the same way as in 5.3, so I will not repeat it here. Also, the language that is recognized by a semi-(n,m)-fsfa is defined in the same way as for an (n,m)-fsfa in Definition 5.4.

8.3 Definition Let Σ be a ranked alphabet. An (n,m)-forest language $L \subseteq \Sigma^{\circledast}[n,m]$ is called *semi-recognizable* if there is a semi-(n,m)-fsfa $\mathfrak{A} = \langle Q, \Sigma, I, F, E\rangle$ with $L(\mathfrak{A}) = L$. The set of all semi-recognizable (n,m)-forest languages over Σ is denoted by $SRec(\Sigma, n, m)$.

8.4 Example Consider again $\Sigma = \Sigma_0 \cup \Sigma_2$ with $\Sigma_0 = \{\alpha\}$ and $\Sigma_2 = \{\sigma\}$ as in Example 5.5. Construct a semi-$(3,2)$-fsfa $\mathfrak{A} = \langle Q, \Sigma, I, F, E\rangle$ as follows.

$$\begin{aligned}
Q &= \{e, o\} \\
I &= \{\langle e, e, e\rangle\} \\
F &= \{\langle e, o\rangle, \langle o, e\rangle\} \\
E_0 &= \{(\langle\rangle, \alpha, o)\} \\
E_2 &= \{(\langle e, e\rangle, \sigma, o), (\langle e, o\rangle, \sigma, e), (\langle o, o\rangle, \sigma, o), (\langle o, e\rangle, \sigma, e)\}
\end{aligned}$$

Then $L(\mathfrak{A}) = \{\langle t_1, t_2\rangle \in \Sigma^{\circledast}[3,2] \mid size(\langle t_1, t_2\rangle) \text{ is odd}\}$. This language is not recognizable (see Examples 6.6 and 6.14).

8.5 Proposition *If Σ is a ranked alphabet, and n and m are natural numbers, then*

$$\begin{aligned}
Rec(\Sigma, n, m) &= SRec(\Sigma, n, m) \quad \textit{for } m = 1 \textit{ and } m = 0, \textit{ and} \\
Rec(\Sigma, n, m) &\subset SRec(\Sigma, n, m) \quad \textit{for every } m \geq 2.
\end{aligned}$$

Proof: It is easy to see from the definitions that every (n,m)-fsfa is also semi-(n,m)-fsfa. Hence, every recognizable forest language is also semi-recognizable. Therefore, $Rec(\Sigma, n, m) \subseteq SRec(\Sigma, n, m)$. For $m = 0$ and $m = 1$, the definitions for $Rec(\Sigma, n, m)$ and $SRec(\Sigma, n, m)$ coincide. (For $m = 0$ see also the proof of Corollary 5.18.) Hence, both sets are equal. To see that the inclusion is proper for $m \geq 2$, it suffices to look at Example 8.4, which shows that the inclusion is proper for $n = 3$ and $m = 2$. Let now $n, m \in \mathbb{N}$ arbitrary with $m \geq 2$. Construct a semi-(n,m)-fsfa $\mathfrak{A} = \langle Q, \Sigma, I, F, E\rangle$ as follows. Let

$$\begin{aligned}
Q &= \{e, o\}, \\
I &= I_1 \times \ldots \times I_n, \text{ where } I_i = \{e\} \text{ for every } i \in \{1, \ldots, n\}, \\
F &= \{\langle e, o, q_3, \ldots, q_m\rangle \mid q_3, \ldots, q_m \in Q\} \cup \{\langle o, e, q_3, \ldots, q_m\rangle \mid q_3, \ldots, q_m \in Q\}, \\
E_0 &= \{(\langle\rangle, \alpha, o)\}, \quad \text{and} \\
E_2 &= \{(\langle e, e\rangle, \sigma, o), (\langle e, o\rangle, \sigma, e), (\langle o, o\rangle, \sigma, o), (\langle o, e\rangle, \sigma, e)\}.
\end{aligned}$$

Then $L(\mathfrak{A}) = \{\langle t_1, t_2, t_3, \ldots, t_m\rangle \in \Sigma^{\circledast}[n,m] \mid size(\langle t_1, t_2\rangle) \text{ is odd}\}$, which is not recognizable. □

8.6 The definition for deterministic (and totally defined) semi-(n,m)-fsfa is identical to Definition 5.9, except for the final state tuples. Therefore, for every $k, l \in \mathbb{N}$, the function

$\delta^{\mathfrak{A}}_{k,l} : Q^k \times \Sigma^{\circledast}[k,l] \to Q^l$ is defined in the same way as in 5.12. Similarly, the definition of the language $L(\mathfrak{A})$ is not different from Definition 5.13. As a consequence, the power set construction of the proof of Proposition 5.17 is unaffected by the change. For this reason, I will just state it again in Proposition 8.8 without repeating the proof.

8.7 Definition Let $n, m \in \mathbb{N}$. A *deterministic, totally defined semi finite-state* (n,m) *forest automation* (*semi-*(n,m)*-dfsfa*) is a tuple $\mathfrak{A} = \langle Q, \Sigma, \vec{q}_I, F, \Delta \rangle$, where Q is a finite set of *states*, Σ is a ranked alphabet, $\vec{q}_I = \langle q_{I1}, \ldots, q_{In} \rangle \in Q^n$ is the *initial state tuple*, $F \subseteq Q^m$ is the set of *final state tuples*, and

$$\Delta = \{\delta_k \mid k \geq 0\}, \quad \text{where} \quad \delta_k : Q^k \times \Sigma_k \to Q \quad (\text{for every } k \geq 0)$$

is the set of *state transition functions*.

8.8 Proposition *Let $\mathfrak{A}$ be a semi-(n,m)-fsfa. Then there exists a semi-(n,m)-dfsfa $\mathfrak{A}'$ such that*

$$L(\mathfrak{A}) = L(\mathfrak{A}').$$

8.9 Proposition *Let Σ be a ranked alphabet and n, m be natural numbers. Furthermore, let $L_1, L_2 \in SRec(\Sigma, n, m)$ be semi-recognizable forest languages. Then*

(i) $L_1 \cup L_2 \in SRec(\Sigma, n, m)$,

(ii) $L_1^C \in SRec(\Sigma, n, m)$, *where* $L_1^C = \Sigma^{\circledast}[n,m] \setminus L_1$,

(iii) $L_1 \cap L_2 \in SRec(\Sigma, n, m)$, *and*

(iv) $L_1 \setminus L_2 \in SRec(\Sigma, n, m)$.

Proof: It is possible, to do the proof of (iii) similar to the proof of the same property for $Rec(\Sigma, n, m)$ (cf. Proposition 6.1). However, it can be done with less effort.

(i) Let $\mathfrak{A}_1 = \langle Q_1, \Sigma, I_1, F_1, E_1 \rangle$ and $\mathfrak{A}_2 = \langle Q_2, \Sigma, I_2, F_2, E_2 \rangle$ be semi-(n,m)-fsfa's (without loss of generality assume $Q_1 \cap Q_2 = \emptyset$) with $L(\mathfrak{A}_1) = L_1$ and $L(\mathfrak{A}_2) = L_2$. For disambiguating the indices, let $I_1 = I_{11} \times \ldots \times I_{1n}$ and $I_2 = I_{21} \times \ldots \times I_{2n}$. Now construct $\mathfrak{A}^{\cup} = \langle Q^{\cup}, \Sigma, I^{\cup}, F^{\cup}, E^{\cup} \rangle$, where

$$\begin{aligned} Q^{\cup} &= Q_1 \cup Q_2, \\ I^{\cup} &= (I_{11} \cup I_{21}) \times \ldots \times (I_{1n} \cup I_{2n}), \\ F^{\cup} &= F_1 \cup F_2, \\ E^{\cup}_k &= E_{1k} \cup E_{2k}. \end{aligned}$$

Then, $\mathfrak{A}^{\cup}$ is a semi-(n,m)-fsfa with $L(\mathfrak{A}^{\cup}) = L(\mathfrak{A}_1) \cup L(\mathfrak{A}_2) = L_1 \cup L_2$.

(ii) Let $\mathfrak{A} = \langle Q, \Sigma, \vec{q}_I, F, \Delta \rangle$ be a semi-(n,m)-dfsfa such that $L(\mathfrak{A}) = L_1$. Let $\mathfrak{A}^C = \langle Q, \Sigma, \vec{q}_I, F^C, \Delta \rangle$ where $F^C = Q^m \setminus F$. Then $L(\mathfrak{A}^C) = \Sigma^{\circledast}[n,m] \setminus L(\mathfrak{A}) = L(\mathfrak{A})^C = L_1^C$.

(iii) $L_1 \cap L_2 = (L_1^C \cup L_2^C)^C$

(iv) $L_1 \setminus L_2 = L_1 \cap L_2^C$

□

8.10 Corollary *For every ranked alphabet Σ and $n, m \in \mathbb{N}$, the set $SRec(\Sigma, n, m)$ forms a boolean algebra, namely a subalgebra of $\mathfrak{P}(\Sigma^{\circledast}[n,m])$. Furthermore, it contains all finite subsets of $\Sigma^{\circledast}[n,m]$.*

8.11 Corollary *Every semi-recognizable language is the union of finitely many recognizable languages. More precisely, for every ranked alphabet* Σ*, every* $n, m \in \mathbb{N}$*, and every* $L \subseteq \Sigma^{\circledast}[n,m]$*, the following are equivalent.*

(i) $L \in SRec(\Sigma^{\circledast}, n, m)$.

(ii) There are an $z \in \mathbb{N}$ *and languages* $L_1, \ldots, L_z \in Rec(\Sigma^{\circledast}, n, m)$ *such that* $L = L_1 \cup \ldots \cup L_z$.

Proof: For the first direction, let $L \in SRec(\Sigma^{\circledast}, n, m)$ and let $\mathfrak{A} = \langle Q, \Sigma, I, F, E \rangle$ be a semi-(n,m)-fsfa with $L(\mathfrak{A}) = L$. Since Q is finite, $F \subseteq Q^m$ is also finite. Hence $F = \{\vec{q}_{F1}, \ldots, \vec{q}_{Fz}\}$ with $z \in \mathbb{N}$ and $\vec{q}_{Fi} \in Q^m$ for every $i = 1, \ldots, z$. Therefore, for every $i = 1, \ldots, z$, the tuple $\mathfrak{A}_i = \langle Q, \Sigma, I, \{\vec{q}_{Fi}\}, E \rangle$ is an (n,m)-fsfa. If we let $L_i = L(\mathfrak{A}_i)$ for every $i = 1, \ldots, z$, then $L = L_1 \cup \ldots \cup L_z$.

For the other direction let $L = L_1 \cup \ldots \cup L_z$ with $L_1, \ldots, L_z \in Rec(\Sigma^{\circledast}, n, m)$. By Proposition 8.5, the languages $L_1, \ldots, L_z \in SRec(\Sigma^{\circledast}, n, m)$. Therefore, $L \in SRec(\Sigma^{\circledast}, n, m)$ by Proposition 8.9. □

8.12 So far, it seems that the class $SRec(\Sigma, n, m)$ has much nicer properties than the class $Rec(\Sigma, n, m)$. However, as mentioned before, a semi-recognizable language L has in general not the form $L = L_1 \times \ldots \times L_m$. And it is also not possible to find for every semi-(n,m)-fsfa $\mathfrak{A}$ a normalized semi-(n,m)-fsfa $\mathfrak{A}'$ such that (10) holds. This is easy to see because whenever a semi-(n,m)-fsfa has only one output state tuple, then it is an (n,m)-fsfa, which (in general) cannot accept a semi-recognizable language. For that reason, I could not find a concatenation operation (nor star operation) under which the semi-recognizable languages are closed.

8.13 Remark Let us define (for every ranked alphabet Σ and natural numbers n, m) the class $SRat(\Sigma^{\circledast}, n, m)$ of *semi-rational languages* similarly to the class $Rat(\Sigma^{\circledast}, n, m)$, with the only difference that the union operation $\cup$ is allowed for every $m \in \mathbb{N}$ (and not only for $m = 1$, cf. Definitions 7.2 and 7.3). Then obviously (by Theorem 7.9 and Corollary 8.11), for every ranked alphabet Σ and natural numbers n, m we have

$$SRec(\Sigma^{\circledast}, n, m) \subseteq SRat(\Sigma^{\circledast}, n, m).$$

This raises the question whether the converse is also true. However, I could neither prove nor disprove it, and it would have gone beyond the scope of this thesis, to put further investigation into this question.

8.14 Conjecture *For every ranked alphabet* Σ *and natural numbers* n, m*,*

$$SRec(\Sigma^{\circledast}, n, m) = SRat(\Sigma^{\circledast}, n, m).$$

8.15 Anyway, the original motivation for introducing the class of semi-recognizable languages is the property stated in Lemma 8.17, which will be needed in the next section.

8.16 Definition Let Σ be a ranked alphabet and let $\sigma \in \Sigma_k$ for some $k \geq 0$. If $L \subseteq \Sigma^{\circledast}[n,1]$ for some $n \geq 0$, then

$$d_\sigma(L) = \{t \in \Sigma^{\circledast}[n,k] \mid t\sigma \in L\}$$

is called the *right quotient* of L with respect to σ.

8.17 Lemma *Let* Σ *be a ranked alphabet, let* $n, k \in \mathbb{N}$*, let* $L \subseteq \Sigma^{\circledast}[n,1]$*, and let* $\sigma \in \Sigma_k$*. If* $L \in SRec(\Sigma, n, 1)$*, then* $d_\sigma(L) \in SRec(\Sigma, n, k)$*.*

Proof: Let $\mathfrak{A} = \langle Q, \Sigma, I, F, E\rangle$ be a semi-$(n,1)$-fsfa with $L(\mathfrak{A}) = L$. Now construct a semi-(n,k)-fsfa $\mathfrak{A}' = \langle Q, \Sigma, I, F', E\rangle$ with

$$F' = \{\langle q_1,\ldots,q_k\rangle \in Q^k \mid \exists q \in F \,.\, (\langle q_1,\ldots,q_k\rangle, \sigma, q) \in E_k\}.$$

Then for every $t \in \Sigma^{\circledast}[n,k]$:

$$\begin{aligned}
t \in L(\mathfrak{A}') &\iff \exists \langle q_1,\ldots,q_k\rangle \in F' \,.\, (\langle I_1,\ldots,I_n\rangle, t, \langle q_1,\ldots,q_k\rangle) \in E^{\mathfrak{A}}_{n,k} \\
&\iff \exists q \in F \,.\, (\langle I_1,\ldots,I_n\rangle, t\sigma, q) \in E^{\mathfrak{A}}_{n,1} \\
&\iff t\sigma \in L(\mathfrak{A}) \\
&\iff t\sigma \in L \\
&\iff t \in d_\sigma(L)
\end{aligned}$$

Hence, $L(\mathfrak{A}') = d_\sigma(L)$. Therefore $d_\sigma(L)$ is semi-recognizable. □

9 Sequential Forest Transducers

In this section, I will introduce the notion of forest transducers. In general, forest transducers can be constructed as straightforward extensions of tree transducers as they can be found in the literature (e.g. [Eng75], [Eng77], [FV98]). In this work, I will restrict myself to the bottom-up case. Furthermore, I will consider only deterministic forest transducers. The motivation is to find a generalization of a theorem by Ginsburg and Rose (cf. [GR66] or [Ber79, p.105]), which gives a "macine independent" characterization for *sequential transducers*, also known as *generalized sequential machines* (*gsm*). For this reason, I propose the name *sequential forest transducer* for a certain class of forest transducers that have a similar "macine independent" characterization as gsm's.

9.1 Definition Let $n \geq 0$ be a natural number. A *deterministic, totally defined, bottom-up n forest transducer* (*n-dbuft*) is a tuple $\mathfrak{B} = \langle Q, \Sigma, \Omega, \vec{q}_I, \Delta, \Lambda\rangle$, where Q is a finite set of *states*, Σ and Ω are ranked alphabets (called *input alphabet* and *output alphabet*, respectively), $\vec{q}_I = \langle q_{I1},\ldots,q_{In}\rangle \in Q^n$ is the *initial state tuple*,

$$\Delta = \{\delta_k \mid k \geq 0\}, \text{ where } \delta_k : Q^k \times \Sigma_k \to Q \text{ for every } k \geq 0$$

is the set of *state transition functions*, and

$$\Lambda = \{\lambda_k \mid k \geq 0\}, \text{ where } \lambda_k : Q^k \times \Sigma_k \to \Omega^{\circledast}[k,1] \text{ for every } k \geq 0$$

is the set of *output functions*.

9.2 Example Let $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$ with $\Sigma_0 = \{\alpha\}$, $\Sigma_1 = \{\gamma\}$, and $\Sigma_2 = \{\sigma\}$. Let $\Omega = \Omega_0 \cup \Omega_1 \cup \Omega_3$ with $\Omega_0 = \{\alpha, \beta\}$, $\Omega_1 = \{\rho\}$, and $\Omega_3 = \{\theta\}$. Now define the 2-dbuft $\mathfrak{B} = \langle Q, \Sigma, \Omega, \vec{q}_I, \Delta, \Lambda\rangle$, where $Q = \{e, o\}$ and $\vec{q}_I = \langle e, e\rangle$. The functions in Δ and Λ are defined as follows.

$$
\begin{array}{rclcrcl}
\delta_0(\langle\rangle,\alpha) & = & o & \qquad & \lambda_0(\langle\rangle,\alpha) & = & \langle\langle\rangle_0\alpha,\langle\rangle_0\alpha,\langle\rangle_0\beta\rangle\theta \\
\delta_1(\langle e\rangle,\gamma) & = & o & & \lambda_1(\langle e\rangle,\gamma) & = & \langle\pi^1_1,\langle\rangle_0\alpha,\langle\rangle_0\beta\rho\rangle\theta \\
\delta_1(\langle o\rangle,\gamma) & = & e & & \lambda_1(\langle o\rangle,\gamma) & = & \langle\pi^1_1\rho,\langle\rangle_0\alpha,\langle\rangle_0\beta\rho\rangle\theta \\
\delta_2(\langle e,e\rangle,\sigma) & = & o & & \lambda_2(\langle e,e\rangle,\sigma) & = & \langle\pi^2_1,\pi^2_2,\langle\rangle_0\beta\rho\rho\rangle\theta \\
\delta_2(\langle e,o\rangle,\sigma) & = & e & & \lambda_2(\langle e,o\rangle,\sigma) & = & \langle\pi^2_1,\pi^2_2\rho,\langle\rangle_0\beta\rho\rho\rangle\theta \\
\delta_2(\langle o,e\rangle,\sigma) & = & e & & \lambda_2(\langle o,e\rangle,\sigma) & = & \langle\pi^2_1\rho,\pi^2_2,\langle\rangle_0\beta\rho\rho\rangle\theta \\
\delta_2(\langle o,o\rangle,\sigma) & = & o & & \lambda_2(\langle o,o\rangle,\sigma) & = & \langle\pi^2_1\rho,\pi^2_2\rho,\langle\rangle_0\beta\rho\rho\rangle\theta
\end{array}
$$

9.3 For a given n-dbuft $\mathfrak{B} = \langle Q,\Sigma,\Omega,\vec{q}_I,\Delta,\Lambda\rangle$, the tuple $\mathfrak{A}_{\mathfrak{B}} = \langle Q,\Sigma,\vec{q}_I,Q,\Delta\rangle$ is an $(n,1)$-dfsfa. This means, that for every $k,l \geq 0$, a function

$$\delta^{\mathfrak{B}}_{k,l} : Q^k \times \Sigma^{\circledast}[k,l] \to Q^l$$

can be defined in the same way as it has been done for (n,m)-dfsfa's in 5.12. For the sake of convenience, let me repeat the equations:

$$
\begin{array}{rcl}
\delta^{\mathfrak{B}}_{k,0}(\langle q_1,\ldots,q_k\rangle,\langle\rangle_k) & = & \langle\rangle, \\
\delta^{\mathfrak{B}}_{k,1}(\langle q_1,\ldots,q_k\rangle,\pi^k_i) & = & q_i \quad (\text{for every } i=1,\ldots,k), \\
\delta^{\mathfrak{B}}_{k,1}(\langle q_1,\ldots,q_k\rangle,t\sigma) & = & \delta_{k'}(\delta^{\mathfrak{B}}_{k,k'}(\langle q_1,\ldots,q_k\rangle,t),\sigma) \\
 & & (\text{where } t\in\Sigma^{\circledast}[k,k'] \text{ and } \sigma\in\Sigma_{k'} \text{ for some } k'\in\mathbb{N}), \\
\delta^{\mathfrak{B}}_{k,l}(\langle q_1,\ldots,q_k\rangle,\langle t_1,\ldots,t_l\rangle) & = & \langle\delta^{\mathfrak{B}}_{k,1}(\langle q_1,\ldots,q_k\rangle,t_1),\ldots,\delta^{\mathfrak{B}}_{k,1}(\langle q_1,\ldots,q_k\rangle,t_l)\rangle \\
 & & (\text{where } l\geq 2 \text{ and } t_1,\ldots,t_l\in\Sigma^{\circledast}[k,1]).
\end{array}
$$

9.4 In a similar way, define for every $k,l \geq 0$ a function

$$\lambda^{\mathfrak{B}}_{k,l} : Q^k \times \Sigma^{\circledast}[k,l] \to \Omega^{\circledast}[k,l]$$

inductively as follows.

$$
\begin{array}{rcl}
\lambda^{\mathfrak{B}}_{k,0}(\langle q_1,\ldots,q_k\rangle,\langle\rangle_k) & = & \langle\rangle_k, \\
\lambda^{\mathfrak{B}}_{k,1}(\langle q_1,\ldots,q_k\rangle,\pi^k_i) & = & \pi^k_i \quad (\text{for every } i=1,\ldots,k), \\
\lambda^{\mathfrak{B}}_{k,1}(\langle q_1,\ldots,q_k\rangle,t\sigma) & = & \lambda^{\mathfrak{B}}_{k,k'}(\langle q_1,\ldots,q_k\rangle,t);\lambda_{k'}(\delta^{\mathfrak{B}}_{k,k'}(\langle q_1,\ldots,q_k\rangle,t),\sigma) \\
 & & (\text{where } t\in\Sigma^{\circledast}[k,k'] \text{ and } \sigma\in\Sigma_{k'} \text{ for some } k'\in\mathbb{N}), \\
\lambda^{\mathfrak{B}}_{k,l}(\langle q_1,\ldots,q_k\rangle,\langle t_1,\ldots,t_l\rangle) & = & \langle\lambda^{\mathfrak{B}}_{k,1}(\langle q_1,\ldots,q_k\rangle,t_1),\ldots,\lambda^{\mathfrak{B}}_{k,1}(\langle q_1,\ldots,q_k\rangle,t_l)\rangle \\
 & & (\text{where } l\geq 2 \text{ and } t_1,\ldots,t_l\in\Sigma^{\circledast}[k,1]).
\end{array}
$$

Those functions are the basis for the description of the transduction that is realized by a transducer.

9.5 Definition Let $\mathfrak{B} = \langle Q,\Sigma,\Omega,\vec{q}_I,\Delta,\Lambda\rangle$ be an n-dbuft. The *transduction realized by* $\mathfrak{B}$ is a family $T^{\mathfrak{B}} = \{\tau^{\mathfrak{B}}_m \mid m \geq 0\}$ of functions

$$\tau^{\mathfrak{B}}_m : \Sigma^{\circledast}[n,m] \to \Omega^{\circledast}[n,m] : t \mapsto \lambda^{\mathfrak{B}}_{n,m}(\vec{q}_I,t).$$

9.6 Lemma *Let $n \in \mathbb{N}$, let $\mathfrak{B} = \langle Q, \Sigma, \Omega, \vec{q}_I, \Delta, \Lambda \rangle$ be an n-dbuft, and let $T^{\mathfrak{B}} = \{\tau_m^{\mathfrak{B}} \mid m \geq 0\}$ be the transduction realized by $\mathfrak{B}$. Then*

$$\begin{array}{rcl}
\tau_0^{\mathfrak{B}}(\langle\rangle_n) & = & \langle\rangle_n, \\
\tau_1^{\mathfrak{B}}(\pi_i^n) & = & \pi_i^n \quad \textit{(for every } i = 1, \ldots, n\textit{)}, \\
\tau_1^{\mathfrak{B}}(t\sigma) & = & \tau_k^{\mathfrak{B}}(t); \lambda_k(\delta_{n,k}^{\mathfrak{B}}(\vec{q}_I, t), \sigma) \quad \textit{(for every } k \geq 0,\ t \in \Sigma^{\circledast}[n,k]\textit{, and } \sigma \in \Sigma_k\textit{)}, \\
\tau_m^{\mathfrak{B}}(\langle t_1, \ldots, t_m \rangle) & = & \langle \tau_1^{\mathfrak{B}}(t_1), \ldots, \tau_1^{\mathfrak{B}}(t_m) \rangle \quad \textit{(for every } m \geq 2 \textit{ and } t_1, \ldots, t_m \in \Sigma^{\circledast}[n,1]\textit{)}.
\end{array}$$

Proof: Immediate from 9.4 and Definition 9.5. □

9.7 Example Consider again the 2-dbuft $\mathfrak{B} = \langle Q, \Sigma, \Omega, \vec{q}_I, \Delta, \Lambda \rangle$ from Example 9.2. Let $t = \langle \langle\rangle_2\alpha, \langle\langle\rangle_2\alpha, \pi_1^2\rangle\sigma, \pi_1^2 \rangle \in \Sigma^{\circledast}[2,3]$, which is shown in Figure 9. Then

$$\begin{array}{rcl}
\tau_3^{\mathfrak{B}}(t) & = & \langle\ \tau_1^{\mathfrak{B}}(\langle\rangle_2\alpha),\ \tau_1^{\mathfrak{B}}(\langle\langle\rangle_2\alpha, \pi_1^2\rangle\sigma),\ \tau_1^{\mathfrak{B}}(\pi_1^2)\ \rangle \\
& = & \langle\ \tau_0^{\mathfrak{B}}(\langle\rangle_2); \lambda_0(\delta_{2,0}^{\mathfrak{B}}(\langle e,e\rangle, \langle\rangle_2), \alpha),\ \tau_2^{\mathfrak{B}}(\langle\langle\rangle_2\alpha, \pi_1^2\rangle); \lambda_2(\delta_{2,2}^{\mathfrak{B}}(\langle e,e\rangle, \langle\langle\rangle_2\alpha, \pi_1^2\rangle), \sigma),\ \pi_1^2\ \rangle \\
& = & \langle\ \langle\rangle_2; \lambda_0(\langle\rangle, \alpha),\ \langle \tau_1^{\mathfrak{B}}(\langle\rangle_2\alpha), \tau_1^{\mathfrak{B}}(\pi_1^2)\rangle; \lambda_2(\langle o,e\rangle, \sigma),\ \pi_1^2\ \rangle \\
& \vdots & \\
& = & \langle\ \langle\langle\rangle_2\alpha, \langle\rangle_2\alpha, \langle\rangle_2\beta\rangle\theta,\ \langle\langle\langle\rangle_2\alpha, \langle\rangle_2\alpha, \langle\rangle_2\beta\rangle\theta\rho, \pi_1^2, \langle\rangle_2\beta\rho\rho\rangle\theta,\ \pi_1^2\ \rangle \ \in\ \Omega^{\circledast}[2,3].
\end{array}$$

9.8 From every n-dbuft, we can construct an equivalent deterministic bottom-up tree transducer (butt) (cf. [Eng75] or [GS97]), where the set of final states $F = Q$. The functions in Δ and Λ of a dbuft $\mathfrak{B} = \langle Q, \Sigma, \Omega, \vec{q}_I, \Delta, \Lambda \rangle$ contain the same information as the term rewriting rules of a butt. Take for instance the dbuft $\mathfrak{B}$ from Example 9.2. Then, construct the butt $B = \langle Q, \Sigma, \Omega, R, Q \rangle$, where

$$\begin{array}{rrcl}
R = \{ & x_1 & \to & e, \\
& x_2 & \to & e, \\
& \alpha & \to & o(\theta(\alpha, \alpha, \beta)), \\
& \gamma(e(x_1)) & \to & o(\theta(x_1, \alpha, \rho(\beta))), \\
& \gamma(o(x_1)) & \to & e(\theta(\rho(x_1), \alpha, \rho(\beta))), \\
& \sigma(e(x_1), e(x_2)) & \to & o(\theta(x_1, x_2, \rho(\rho(\beta)))), \\
& \sigma(e(x_1), o(x_2)) & \to & e(\theta(x_1, \rho(x_2), \rho(\rho(\beta)))), \\
& \sigma(o(x_1), e(x_2)) & \to & e(\theta(\rho(x_1), x_2, \rho(\rho(\beta)))), \\
& \sigma(o(x_1), o(x_2)) & \to & o(\theta(\rho(x_1), \rho(x_2), \rho(\rho(\beta)))) \ \}
\end{array}$$

is the set of rules. Since the butt is deterministic and totally defined, it induces a function $f_B : T_\Sigma(X_2) \to T_\Omega(X_2)$. If $T^{\mathfrak{B}} = \{\tau_m^{\mathfrak{B}} \mid m \geq 0\}$ is the transduction realized by $\mathfrak{B}$, then we can take the bijections $\varphi_\Sigma : \Sigma^{\circledast}[2,1] \to T_\Sigma(X_2)$ and $\varphi_\Omega : \Omega^{\circledast}[2,1] \to T_\Omega(X_2)$ (by Proposition 4.27) to obtain the following connection between the dbuft $\mathfrak{B}$ and the butt B. For every $t \in \Sigma^{\circledast}[2,1]$,

$$\varphi_\Omega(\tau_1^{\mathfrak{B}}(t)) \quad = \quad f_B(\varphi_\Sigma(t)). \tag{22}$$

This means, that several results from the theory of tree transducers can be applied to forest transducers.

9.9 Lemma *Let $n \in \mathbb{N}$, let $\mathfrak{B} = \langle Q, \Sigma, \Omega, \vec{q}_I, \Delta, \Lambda \rangle$ be an n-dbuft, and let $T^{\mathfrak{B}} = \{\tau_m^{\mathfrak{B}} \mid m \geq 0\}$ be the transduction realized by $\mathfrak{B}$. Then, for every $m \in \mathbb{N}$ and $L \subseteq \Omega^{\circledast}[n, m]$, if $L \in Rec(\Omega^{\circledast}, n, m)$, then $(\tau_m^{\mathfrak{B}})^{-1}(L) \in Rec(\Sigma^{\circledast}, n, m)$.*

Proof: This lemma restates a well-known property of deterministic butt's. In Proposition 20.1.(iii) in [GS97, p.45], it is stated that for every recognizable tree language $L \subseteq T_\Omega(Y)$ and every bottom-up tree transformation $\tau \subseteq T_\Sigma(X) \times T_\Omega(Y)$, the tree language $\tau^{-1}(L) \subseteq T_\Sigma(X)$ is recognizable. By 9.8, an n-dbuft $\mathfrak{B} = \langle Q, \Sigma, \Omega, \vec{q}_I, \Delta, \Lambda \rangle$ can be seen as a deterministic butt inducing a function $f : T_\Sigma(X_n) \to T_\Omega(X_n)$. The graph of this function is a special case of a bottom-up tree transformation. Hence, by (22), for every language $L \in Rec(\Omega^{\circledast}, n, 1)$, we have $(\tau_1^{\mathfrak{B}})^{-1}(L) \in Rec(\Sigma^{\circledast}, n, 1)$. Since, by Lemma 9.6, for every $t_1, \ldots, t_m \in \Sigma^{\circledast}[n, 1]$, we have $\tau_m^{\mathfrak{B}}(\langle t_1, \ldots, t_m \rangle) = \langle \tau_1^{\mathfrak{B}}(t_1), \ldots, \tau_1^{\mathfrak{B}}(t_m) \rangle$, it follows by Proposition 6.12 that for every $m \in \mathbb{N}$ and $L \subseteq \Omega^{\circledast}[n, m]$, if $L \in Rec(\Omega^{\circledast}, n, m)$, then $(\tau_m^{\mathfrak{B}})^{-1}(L) \in Rec(\Sigma^{\circledast}, n, m)$. □

9.10 As mentioned before, in this section, I will generalize a theorem by Ginsburg and Rose (cf. [GR66]), that gives a "machine independent" characterization for sequential transducers (see also [Ber79, ch.IV]). The original proof contains two constructions on word languages, that I could not extend to tree languages. For that reason, I introduced the concept of forests.
Before I will give the definition of a sequential forest transducer and the "machine independent" characterization for the functions realized by them, I will define the class of sequential forest relabelings, which is a subclass of the class of sequential forest transducers. The motivation is to separate the two constructions mentioned above. I believe that this will help to understand why the proof works.

9.11 The name sequential forest relabeling is motivated by the fact that the function realized by such a relabeling is simply a relabeling of the nodes of the trees in the input forest. For the formal definition, let me define for every ranked alphabet Σ the set

$$\Sigma^{\odot} = \{\varepsilon_k \sigma \mid k \geq 0 \land \sigma \in \Sigma_k\} \subseteq \Sigma^{\circledast}.$$

The set $\Sigma^{\odot}$ is an $\mathbb{N}$-indexed set with $\Sigma_k^{\odot} = \{\varepsilon_k \sigma \mid \sigma \in \Sigma_k\} \subseteq \Sigma^{\circledast}[k, 1]$ for every $k \geq 0$.

9.12 Definition Let $n \in \mathbb{N}$ and $\mathfrak{R} = \langle Q, \Sigma, \Omega, \vec{q}_I, \Delta, \Lambda \rangle$ be an n-dbuft. Then $\mathfrak{R}$ is called a *sequential n forest relabeling* (n-*seqrel*) if for every $k \geq 0$, every $\sigma \in \Sigma_k$, and every $q_1, \ldots, q_k \in Q$,

$$\lambda_k(\langle q_1, \ldots, q_k \rangle, \sigma) \in \Omega_k^{\odot}.$$

9.13 Theorem *Let Σ and Ω be ranked alphabets, and let n be a natural number. Let $T = \{\tau_m \mid m \geq 0\}$ be a family of functions $\tau_m : \Sigma^{\circledast}[n, m] \to \Omega^{\circledast}[n, m]$. Then T is a transduction realized by an n-seqrel if and only if T satisfies the following five conditions.*

(i) For every $i = 1, \ldots, n$,

$$\tau_1(\pi_i^n) = \pi_i^n.$$

(ii) For every $k \in \mathbb{N}$, every $t \in \Sigma^{\circledast}[n, k]$, and every $\sigma \in \Sigma_k$, there is a $\nu \in \Omega_k^{\odot}$ such that

$$\tau_1(t\sigma) = \tau_k(t); \nu.$$

(iii) For every $k \in \mathbb{N}$, every $t \in \Sigma^{\circledast}[n,k]$, and every $\sigma \in \Sigma_k$,

$$size(\tau_1(t\sigma)) - size(\tau_k(t)) = 1.$$

(iv) For every $m \in \mathbb{N}$, if $L \in Rat(\Omega, n, m)$, then $\tau_m^{-1}(L) \in Rat(\Sigma, n, m)$.

(v) For every $m \geq 0$ and $t_1, \ldots, t_m \in \Sigma^{\circledast}[n,1]$,

$$\tau_m(\langle t_1, \ldots, t_m \rangle) = \langle \tau_1(t_1), \ldots, \tau_1(t_m) \rangle.$$

Proof: Let me first assume, that T is realized by an n-seqrel $\mathfrak{R} = \langle Q, \Sigma, \Omega, \vec{q}_I, \Delta, \Lambda \rangle$, with $T^{\mathfrak{R}} = T$. Now I have to show that $T^{\mathfrak{R}}$ fulfills Conditions (i)-(v) of the theorem.

(i) For every $i = 1, \ldots, n$, we have $\tau_1^{\mathfrak{R}}(\pi_i^n) = \pi_i^n$ by Lemma 9.6.

(ii) For every $k \geq 0$, $t \in \Sigma^{\circledast}[n,k]$, $\sigma \in \Sigma_k$, let $\nu = \lambda_k(\delta_{n,k}^{\mathfrak{R}}(\vec{q}_I, t), \sigma)$. Then $\tau_1^{\mathfrak{R}}(t\sigma) = \tau_k^{\mathfrak{R}}(t); \nu$ as required (again by Lemma 9.6).

(iii) Let $\rho \in \Omega_k$ such that $\nu = \varepsilon_k \rho$. Then

$$size(\tau_1^{\mathfrak{R}}(t\sigma)) = size(\tau_k^{\mathfrak{R}}(t); \nu) = size(\tau_k^{\mathfrak{R}}(t); \varepsilon_k \rho) = size(\tau_k^{\mathfrak{R}}(t)) + 1.$$

(iv) Immediate from Lemma 9.9 and Theorem 7.9.

(v) Immediate from Lemma 9.6.

For the other direction, assume that $T = \{\tau_m \mid m \in \mathbb{N}\}$ is a family of functions $\tau_m : \Sigma^{\circledast}[n,m] \to \Omega^{\circledast}[n,m]$ that satisfies Conditions (i) to (v) of Theorem 9.13. Then, Condition (ii) induces a function

$$f_k : \Sigma^{\circledast}[n,k] \times \Sigma_k \to \Omega_k^{\odot}$$

for every $k \geq 0$ by $f_k(t, \sigma) = \nu$ for every $t \in \Sigma^{\circledast}[n,k]$ and $\sigma \in \Sigma_k$. By currying, we can obtain functions

$$\hat{f}_k : \Sigma^{\circledast}[n,k] \to [\Sigma_k \to \Omega_k^{\odot}],$$

where $[\Sigma_k \to \Omega_k^{\odot}]$ denotes the set of all functions from Σ_k to $\Omega_k^{\odot}$. Now define for every $k \in \mathbb{N}$ the set

$$H_k = im(\hat{f}_k) = \{\eta \in [\Sigma_k \to \Omega_k^{\odot}] \mid \exists t \in \Sigma^{\circledast}[n,k] \,.\, \hat{f}_k(t) = \eta\}$$

to be the image of the function $\hat{f}_k$. Then, for every $k \geq 0$, the set H_k is finite because the image of every function $\eta \in H_k$ is restricted to a finite set (namely a subset of $\Omega_k^{\odot}$). Now, for every $k \geq 0$ and every $\eta \in H_k$ define a language $L_k(\eta) \subseteq \Sigma^{\circledast}[n,k]$ as follows.

$$L_k(\eta) = \{t \in \Sigma^{\circledast}[n,k] \mid \hat{f}_k(t) = \eta\}.$$

Obviously, for every $k \geq 0$ there is only a finite number of such languages because H_k is finite. Furthermore, it holds the following claim.

Claim 1: For every $k \geq 0$, The set $\Sigma^{\circledast}[n,k]$ is the disjoint union of all $L_k(\eta)$ with $\eta \in H_k$, i.e.

$$\Sigma^{\circledast}[n,k] = \bigcup_{\eta \in H_k} L_k(\eta) \text{ and for every } \eta_1, \eta_2 \in H_k \text{ with } \eta_1 \neq \eta_2,\ L_k(\eta_1) \cap L_k(\eta_2) = \emptyset.$$

Proof of Claim 1: Consider the equivalence relation $\sim_k \subseteq \Sigma^{\circledast}[n,k] \times \Sigma^{\circledast}[n,k]$ induced by $\hat{f}_k$:

$$t \sim_k t' \iff \hat{f}_k(t) = \hat{f}_k(t').$$

Then, for every $\eta \in H_k$, the set $L_k(\eta)$ represents one equivalence class. And for every equivalence relation, the carrier set is the disjoint union of all equivalence classes. From this, Claim 1 follows immediately.

In the next step, I will show that for every $k \geq 0$ and $\eta \in H_k$, the language $L_k(\eta) \subseteq \Sigma^{\circledast}[n,k]$ is semi-recognizable. First, for every $k \geq 0$, $\sigma \in \Sigma_k$, and $\nu \in \Omega_k^{\odot}$, define the language

$$N_k(\sigma,\nu) = \{t \in \Sigma^{\circledast}[n,k] \mid \tau_1(t\sigma) = \tau_k(t);\nu\}.$$

Claim 2: $N_k(\sigma,\nu) \in \mathit{SRec}(\Sigma,n,k)$.

Proof of Claim 2: Consider $N_k'(\sigma,\nu) = d_\sigma(\tau_1^{-1}(\Omega^{\circledast}[n,k];\nu))$.
Since $\Omega^{\circledast}[n,k] \in \mathit{Rec}(\Omega,n,k)$, we have $\Omega^{\circledast}[n,k];\nu \in \mathit{Rec}(\Omega,n,1)$ by Lemma 6.31 and Proposition 6.34. Hence, $\tau_1^{-1}(\Omega^{\circledast}[n,k];\nu) \in \mathit{Rec}(\Sigma,n,1)$ by Theorem 7.9 and Condition (iv). By Lemma 8.17, it follows that $d_\sigma(\tau_1^{-1}(\Omega^{\circledast}[n,k];\nu)) \in \mathit{SRec}(\Sigma,n,k)$. Therefore, $N_k'(\sigma,\nu)$ is semi-recognizable.
Let me now show that $N_k(\sigma,\nu) = N_k'(\sigma,\nu)$. Suppose $t \in N_k(\sigma,\nu)$. Then $\tau_1(t\sigma) = \tau_k(t);\nu$. Hence $\tau_1(t\sigma) \in \Omega^{\circledast}[n,k];\nu$, and therefore $t\sigma \in \tau_1^{-1}(\Omega^{\circledast}[n,k];\nu)$. Then by definition of d_σ, we have $t \in d_\sigma(\tau_1^{-1}(\Omega^{\circledast}[n,k];\nu))$, which means that $t \in N_k'(\sigma,\nu)$.
Conversely, suppose $t \in N_k'(\sigma,\nu)$. This means that $t\sigma \in \tau_1^{-1}(\Omega^{\circledast}[n,k];\nu)$. Therefore $\tau_1(t\sigma) \in \Omega^{\circledast}[n,k];\nu$. But by Condition (ii), there is a $\nu' \in \Omega_k^{\odot}$ with $\tau_1(t\sigma) = \tau_k(t);\nu'$. Since τ_1 is a total function, $\tau_k(t);\nu' \in \Omega^{\circledast}[n,k];\nu$. Since $\nu = \varepsilon_k\rho$ and $\nu' = \varepsilon_k\rho'$ for some $\rho,\rho' \in \Omega_k$, we get $\rho = \rho'$ and therefore, $\nu = \nu'$, which completes the proof of Claim 2.

Let me now come back to the languages $L_k(\eta)$ as defined above. For every $k \geq 0$, $\eta \in H_k$, and $t \in \Sigma^{\circledast}[n,k]$, it holds that

$$\begin{aligned} t \in L_k(\eta) \quad &\Longleftrightarrow \quad \forall\sigma \in \Sigma_k \;.\; \tau_1(t\sigma) = \tau_k(t);\eta(\sigma) \\ &\Longleftrightarrow \quad \forall\sigma \in \Sigma_k \;.\; t \in N_k(\sigma,\eta(\sigma)). \end{aligned}$$

Therefore, for every $k \geq 0$ and $\eta \in H_k$,

$$L_k(\eta) = \bigcap_{\sigma\in\Sigma_k} N_k(\sigma,\eta(\sigma)).$$

Since semi-recognizable languages are closed under intersection, this proves that $L_k(\eta) \in \mathit{SRec}(\Sigma,n,k)$.

Now let

$$H = \bigcup_{k\geq 0} H_k.$$

Then H is finite because H_k is finite for every $k \geq 0$, and for almost all $k \geq 0$, the set $H_k = \emptyset$ (cf. Remark 3.4). Therefore, we can assume without loss of generality that $H = \{\eta_1,\ldots,\eta_r\}$ for some $r \in \mathbb{N}$. Furthermore, define a function $\mathit{rank} : H \to \mathbb{N}$ by $\mathit{rank}(\eta) = k$ if $\eta \in H_k$. (Remark: H is an $\mathbb{N}$-indexed set.) For every $i = 1,\ldots,r$, define now the set $L(\eta_i) = L_{\mathit{rank}(\eta_i)}(\eta_i)$. Then $L(\eta_i) \in \mathit{SRec}(\Sigma,n,\mathit{rank}(\eta_i))$.

Hence, for every $i = 1,\ldots,r$ there is a semi-$(n,\mathit{rank}(\eta_i))$-dfsfa $\mathfrak{A}_i = \langle Q_i,\Sigma,\vec{q}_{I\,i},F_i,\Delta_i\rangle$ (with $\vec{q}_{I\,i} = \langle q_{I\,i1},\ldots,q_{I\,in}\rangle$ and $F_i \subseteq Q^{\mathit{rank}(\eta_i)}$) such that $L(\mathfrak{A}_i) = L(\eta_i)$. From those automata, I will now construct an n-seqrel $\mathfrak{R} = \langle Q,\Sigma,\Omega,\vec{q}_I,\Delta,\Lambda\rangle$ such that $T^{\mathfrak{R}} = T$. Let

$$\begin{aligned} Q &= Q_1 \times Q_2 \times \ldots \times Q_r, \\ \vec{q}_I &= \langle\langle q_{I\,11},\ldots,q_{I\,r1}\rangle,\ldots,\langle q_{I\,1n},\ldots,q_{I\,rn}\rangle\rangle, \\ \delta_k(\langle\langle q_{11},\ldots,q_{r1}\rangle,\ldots,\langle q_{1k},\ldots,q_{rk}\rangle\rangle,\sigma) &= \langle\delta_{1k}(\langle q_{11},\ldots,q_{1k}\rangle,\sigma),\ldots,\delta_{rk}(\langle q_{r1},\ldots,q_{rk}\rangle,\sigma)\rangle. \end{aligned}$$

Informally speaking, this says that the underlying Σ-algebra of $\mathfrak{R}$ is the cartesian product of the underlying Σ-algebras of the automata $\mathfrak{A}_1,\ldots,\mathfrak{A}_r$ (cf. 5.15). It immediately follows (by the same induction argument as in the proof for Proposition 6.1) that for every $k \geq 0$, every $t \in \Sigma^{\circledast}[k,1]$, every $q_{11},\ldots,q_{1k} \in Q_1$, ..., and every $q_{r1},\ldots,q_{rk} \in Q_r$,

$$\delta_{k,1}^{\mathfrak{R}}(\langle\langle q_{11},\ldots,q_{r1}\rangle,\ldots,\langle q_{1k},\ldots,q_{rk}\rangle\rangle,t) = \langle\delta_{k,1}^{\mathfrak{A}_1}(\langle q_{11},\ldots,q_{1k}\rangle,t),\ldots,\delta_{k,1}^{\mathfrak{A}_r}(\langle q_{r1},\ldots,q_{rk}\rangle,t)\rangle.$$

It remains to define $\Lambda = \{\lambda_k \mid k \geq 0\}$. For every $k \geq 0$, every $q_{11}, \dots, q_{1k} \in Q_1$, ..., every $q_{r1}, \dots, q_{rk} \in Q_r$, and every $\sigma \in \Sigma_k$, let

$$\lambda_k(\langle\langle q_{11}, \dots, q_{r1}\rangle, \dots, \langle q_{1k}, \dots, q_{rk}\rangle\rangle, \sigma) \quad = \quad \eta_i(\sigma), \tag{23}$$

where i is chosen such that

$$\begin{array}{rcl} \eta_i & \in & H_k, \\ \langle q_{i1}, \dots, q_{ik}\rangle & \in & F_i, \text{ and} \\ \langle q_{j1}, \dots, q_{jk}\rangle & \notin & F_j \text{ for every } j \neq i \text{ with } \eta_j \in H_k. \end{array}$$

If there is no such i (this case can occur only if one of the states $\langle q_{11}, \dots, q_{r1}\rangle, \dots, \langle q_{1k}, \dots, q_{rk}\rangle \in Q$ is inaccessible), then pick any $\eta_i \in H_k$.

In order to complete the proof, it remains to show that indeed $T^{\mathfrak{R}} = T$, i.e. for every $m \geq 0$ and $t \in \Sigma^{\circledast}[n, m]$, it holds that $\tau_m^{\mathfrak{R}}(t) = \tau_m(t)$. This will be done by induction on the structure of $t \in \Sigma^{\circledast}[n, m]$.

(i) $\underline{t = \langle\rangle_n}$. Then

$$\tau_1(\langle\rangle_n) = \langle\rangle_n = \tau_1^{\mathfrak{R}}(\langle\rangle_n).$$

(ii) $\underline{t = \pi_i^n}$ for some $i = 1, \dots, n$. Then

$$\begin{array}{rcl} \tau_1(\pi_i^n) & = & \pi_i^n \text{ (by Condition (i))} \\ & = & \tau_1^{\mathfrak{R}}(\pi_i^n) \text{ (by Lemma 9.6)} \end{array}$$

(iii) $\underline{t = t'\sigma}$ for some $k \geq 0$, some $t' \in \Sigma^{\circledast}[n, k]$, and some $\sigma \in \Sigma_k$. Then

$$\begin{array}{rcll} \tau_1(t) & = & \tau_1(t'\sigma) & \\ & = & \tau_k(t'); f_k(t', \sigma) & \text{(by definitions of } T \text{ and } f_k) \\ & = & \tau_k(t'); \hat{f}_k(t')(\sigma) & \text{(by definition of } \hat{f}_k) \\ & = & \tau_k^{\mathfrak{R}}(t'); \hat{f}_k(t')(\sigma) & \text{(by induction hypothesis)} \end{array}$$

We know that $\hat{f}_k(t') = \eta_i$ for exactly one $i \in \{1, \dots, r\}$. In particular, $\eta_i \in H_k$. This means that $t' \in L(\mathfrak{A}_i)$ and $t' \notin L(\mathfrak{A}_j)$ for every $j \neq i$ with $\eta_j \in H_k$. Hence, $\delta_{n,k}^{\mathfrak{A}_i}(\langle q_{I\,i1}, \dots, q_{I\,in}\rangle, t') \in F_i$ and $\delta_{n,k}^{\mathfrak{A}_j}(\langle q_{I\,j1}, \dots, q_{I\,jn}\rangle, t') \notin F_j$ for every $j \neq i$ (for $\eta_j \in H_k$ as well as $\eta_j \notin H_k$). Therefore,

$$\lambda_k(\delta_{n,k}^{\mathfrak{R}}(\langle\langle q_{I\,11}, \dots, q_{I\,r1}\rangle, \dots, \langle q_{I\,1n}, \dots, q_{I\,rn}\rangle\rangle, \sigma) = \eta_i(\sigma).$$

Now we can continue the equation sequence from above.

$$\begin{array}{rcl} \tau_1(t) & = & \tau_k^{\mathfrak{R}}(t'); \hat{f}_k(t')(\sigma) \\ & = & \tau_k^{\mathfrak{R}}(t'); \eta_i(\sigma) \\ & = & \tau_k^{\mathfrak{R}}(t'); \lambda_k(\delta_{n,k}^{\mathfrak{R}}(\vec{q}_I, t'), \sigma) \\ & = & \tau_1^{\mathfrak{R}}(t'\sigma) \quad \text{(by definition of } \tau^{\mathfrak{R}}) \\ & = & \tau_1^{\mathfrak{R}}(t) \end{array}$$

(iv) $\underline{t = \langle t_1, \dots, t_m\rangle}$ for some $m \geq 2$ and some $t_1, \dots, t_m \in \Sigma^{\circledast}[n, 1]$. Then

$$\begin{array}{rcl} \tau_m(t) & = & \langle \tau_1(t_1), \dots, \tau_1(t_m)\rangle \text{ (by Condition (v))} \\ & = & \langle \tau_1^{\mathfrak{R}}(t_1), \dots, \tau_1^{\mathfrak{R}}(t_m)\rangle \text{ (by induction hypothesis)} \\ & = & \tau_m^{\mathfrak{R}}(t) \text{ (by Lemma 9.6)} \end{array}$$

This completes the proof of the theorem. □

9.14 Remark In 9.10, I mentioned that there are two constructions that make the generalization of the result in [GR66] difficult. The first one is to show that for every $k \in \mathbb{N}$ and $\eta \in H_k$, the language $L_k(\eta)$ is semi-recognizable—for sequential forest relabelings, this is rather trivial—and the second one is the construction of the n-seqrel from the automata for the languages $L_k(\eta)$. In order to ease the understanding of that construction, I will give a simple example.

9.15 Example Let $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$ with $\Sigma_0 = \{\alpha, \beta\}$, $\Sigma_1 = \{\gamma\}$, and $\Sigma_2 = \{\sigma\}$. Let $\Omega = \Omega_0 \cup \Omega_1 \cup \Omega_2$ with $\Omega_0 = \{\alpha', \beta'\}$, $\Omega_1 = \{\gamma'_1, \gamma'_2, \gamma'_3\}$, and $\Omega_2 = \{\sigma'\}$. In order to simplify things, let $n = 0$. Now, let $T = \{\tau_m \mid m \geq 0\}$ be a family of functions $\tau_m : \Sigma^{\circledast}[0,m] \to \Omega^{\circledast}[0,m]$ such that Condition (v) of Theorem 9.13 is satisfied, and such that

$$
\begin{aligned}
\tau_1(\langle\rangle_0\alpha) &= \tau_0(\langle\rangle_0);\varepsilon_0\alpha' \;=\; \langle\rangle_0\alpha', \\
\tau_1(\langle\rangle_0\beta) &= \tau_0(\langle\rangle_0);\varepsilon_0\beta' \;=\; \langle\rangle_0\beta', \\
\tau_1(t\gamma) &= \begin{cases} \tau_1(t)\gamma'_1 & \text{if } \mathit{size}(t) \text{ is odd} \\ \tau_1(t)\gamma'_2 & \text{if } \mathit{size}(t) \text{ is even} \end{cases} \\
\tau_1(t\sigma) &= \tau_2(t);\varepsilon_2\sigma' \;=\; \tau_2(t)\sigma'.
\end{aligned}
$$

(Remark: The symbol $\gamma'_3 \in \Omega_1$ does not occur in the output.) Observe that T does satisfy Conditions (ii), (iii) and (iv) (of Theorem 9.13). Condition (i) is vacuous because $n = 0$. Let me now carry out the construction of the proof of Theorem 9.13 for this example. The set H_0 contains only one function η_1, which maps α to $\varepsilon_0\alpha'$, and β to $\varepsilon_0\beta'$. (Remark: The set $[\Sigma_0 \to \Omega_0^{\odot}]$ contains four elements in this example.) The set H_1 contains two functions η_2 and η_3, mapping γ to $\varepsilon_1\gamma'_1$ and $\varepsilon_1\gamma'_2$, respectively. (Remark: The set $[\Sigma_1 \to \Omega_1^{\odot}]$ contains a third function that maps γ to $\varepsilon_1\gamma'_3$, but which does not occur in H_1.) The set H_2 contains only one function η_4, which maps σ to σ'. Therefore, the set $H = H_0 \cup H_1 \cup H_2 = \{\eta_1, \eta_2, \eta_3, \eta_4\}$. contains four elements. The corresponding languages are

$$
\begin{aligned}
L(\eta_1) &= \Sigma^{\circledast}[0,0] = \{\langle\rangle_0\} \\
L(\eta_2) &= \{t \in \Sigma^{\circledast}[0,1] \mid \mathit{size}(t) \text{ is odd}\} \\
L(\eta_3) &= \{t \in \Sigma^{\circledast}[0,1] \mid \mathit{size}(t) \text{ is even}\} \\
L(\eta_4) &= \Sigma^{\circledast}[0,2]
\end{aligned}
$$

Note: It is easy to see that the statement of Claim 1 does hold.
All four languages are indeed semi-recognizable. (In this example, they are also recognizable, but in general this is not the case.) Hence, we obtain four automata $\mathfrak{A}_1$, $\mathfrak{A}_2$, $\mathfrak{A}_3$, and $\mathfrak{A}_4$, which are (in this example) so trivial that they can be spelled out.

- $\mathfrak{A}_1 = \langle Q_1, \Sigma, \vec{q}_{I1}, F_1, \Delta_1\rangle$ with
  - $Q_1 = \{a\}$,
  - $F_1 = \{\langle\rangle\}$,
  - $\vec{q}_{I1} = \langle\rangle$,
  - There is only one possibility for the functions in $\Delta_1 = \{\delta_{1k} \mid k \geq 0\}$ because Q is a singleton.

$$
\begin{array}{rclrcl}
\delta_{10}(\langle\rangle,\alpha) & = & a & \delta_{11}(\langle a\rangle,\gamma) & = & a \\
\delta_{10}(\langle\rangle,\beta) & = & a & \delta_{12}(\langle a,a\rangle,\sigma) & = & a
\end{array}
$$

- $\mathfrak{A}_2 = \langle Q_2, \Sigma, \vec{q}_{I2}, F_2, \Delta_2\rangle$ with
  - $Q_2 = \{e, o\}$,
  - $F_2 = \{o\}$,
  - $\vec{q}_{I2} = \langle\rangle$,
  - The functions in $\Delta_2 = \{\delta_{2k} \mid k \geq 0\}$ are similar to those of example 5.14.

$$
\begin{array}{rclrcl}
\delta_{20}(\langle\rangle,\alpha) & = & o & \delta_{22}(\langle e,e\rangle,\sigma) & = & o \\
\delta_{20}(\langle\rangle,\beta) & = & o & \delta_{22}(\langle e,o\rangle,\sigma) & = & e \\
\delta_{21}(\langle e\rangle,\gamma) & = & o & \delta_{22}(\langle o,e\rangle,\sigma) & = & e \\
\delta_{21}(\langle o\rangle,\gamma) & = & e & \delta_{22}(\langle o,o\rangle,\sigma) & = & o
\end{array}
$$

- $\mathfrak{A}_3 = \langle Q_3, \Sigma, \vec{q}_{I3}, F_3, \Delta_3\rangle$ differs from $\mathfrak{A}_2$ only in the final state.
  - $Q_3 = \{e, o\}$,
  - $F_3 = \{e\}$,
  - $\vec{q}_{I3} = \langle\rangle$,
  - The functions in $\Delta_3 = \{\delta_{3k} \mid k \geq 0\}$ are the same as in Δ_2.

$$
\begin{array}{rclrcl}
\delta_{30}(\langle\rangle,\alpha) & = & o & \delta_{32}(\langle e,e\rangle,\sigma) & = & o \\
\delta_{30}(\langle\rangle,\beta) & = & o & \delta_{32}(\langle e,o\rangle,\sigma) & = & e \\
\delta_{31}(\langle e\rangle,\gamma) & = & o & \delta_{32}(\langle o,e\rangle,\sigma) & = & e \\
\delta_{31}(\langle o\rangle,\gamma) & = & e & \delta_{32}(\langle o,o\rangle,\sigma) & = & o
\end{array}
$$

- $\mathfrak{A}_4 = \langle Q_4, \Sigma, \vec{q}_{I4}, F_4, \Delta_4\rangle$ with
  - $Q_4 = \{a\}$,
  - $F_4 = \{\langle a, a\rangle\}$,
  - $\vec{q}_{I4} = \langle\rangle$,
  - The functions in $\Delta_4 = \{\delta_{4k} \mid k \geq 0\}$ are the same as in Δ_1.

$$
\begin{array}{rclrcl}
\delta_{40}(\langle\rangle,\alpha) & = & a & \delta_{41}(\langle a\rangle,\gamma) & = & a \\
\delta_{40}(\langle\rangle,\beta) & = & a & \delta_{42}(\langle a,a\rangle,\sigma) & = & a
\end{array}
$$

From those automata, we can construct the sequential relabeling $\mathfrak{R} = \langle Q, \Sigma, \Omega, \vec{q}_I, \Delta, \Lambda\rangle$ such that $T^{\mathfrak{R}} = T$ according to the general construction in the proof:

$$
\begin{array}{rcl}
Q & = & Q_1 \times Q_2 \times Q_3 \times Q_4 \ = \ \{\langle a,e,e,a\rangle, \langle a,e,o,a\rangle, \langle a,o,e,a\rangle, \langle a,o,o,a\rangle\} \\
\vec{q}_I & = & \langle\rangle
\end{array}
$$

The definition of $\Delta = \{\delta_k \mid k \geq 0\}$ is now straightforward. For the sake of completeness, I will spell out every single case.

$$
\begin{array}{rclrcl}
\delta_0(\langle\rangle, \alpha) & = & \langle a,o,o,a\rangle & \delta_2(\langle\langle a,e,o,a\rangle,\langle a,e,o,a\rangle\rangle,\sigma) & = & \langle a,o,o,a\rangle \\
\delta_0(\langle\rangle, \beta) & = & \langle a,o,o,a\rangle & \delta_2(\langle\langle a,e,o,a\rangle,\langle a,o,e,a\rangle\rangle,\sigma) & = & \langle a,e,e,a\rangle \\
\delta_1(\langle\langle a,e,e,a\rangle\rangle,\gamma) & = & \langle a,o,o,a\rangle & \delta_2(\langle\langle a,e,o,a\rangle,\langle a,o,o,a\rangle\rangle,\sigma) & = & \langle a,e,o,a\rangle \\
\delta_1(\langle\langle a,e,o,a\rangle\rangle,\gamma) & = & \langle a,o,e,a\rangle & \delta_2(\langle\langle a,o,e,a\rangle,\langle a,e,e,a\rangle\rangle,\sigma) & = & \langle a,e,o,a\rangle \\
\delta_1(\langle\langle a,o,e,a\rangle\rangle,\gamma) & = & \langle a,e,o,a\rangle & \delta_2(\langle\langle a,o,e,a\rangle,\langle a,e,o,a\rangle\rangle,\sigma) & = & \langle a,e,e,a\rangle \\
\delta_1(\langle\langle a,o,o,a\rangle\rangle,\gamma) & = & \langle a,e,e,a\rangle & \delta_2(\langle\langle a,o,e,a\rangle,\langle a,o,e,a\rangle\rangle,\sigma) & = & \langle a,o,o,a\rangle \\
\delta_2(\langle\langle a,e,e,a\rangle,\langle a,e,e,a\rangle\rangle,\sigma) & = & \langle a,o,o,a\rangle & \delta_2(\langle\langle a,o,e,a\rangle,\langle a,o,o,a\rangle\rangle,\sigma) & = & \langle a,o,e,a\rangle \\
\delta_2(\langle\langle a,e,e,a\rangle,\langle a,e,o,a\rangle\rangle,\sigma) & = & \langle a,o,e,a\rangle & \delta_2(\langle\langle a,o,o,a\rangle,\langle a,e,e,a\rangle\rangle,\sigma) & = & \langle a,e,e,a\rangle \\
\delta_2(\langle\langle a,e,e,a\rangle,\langle a,o,e,a\rangle\rangle,\sigma) & = & \langle a,e,o,a\rangle & \delta_2(\langle\langle a,o,o,a\rangle,\langle a,e,o,a\rangle\rangle,\sigma) & = & \langle a,e,o,a\rangle \\
\delta_2(\langle\langle a,e,e,a\rangle,\langle a,o,o,a\rangle\rangle,\sigma) & = & \langle a,e,e,a\rangle & \delta_2(\langle\langle a,o,o,a\rangle,\langle a,o,e,a\rangle\rangle,\sigma) & = & \langle a,o,e,a\rangle \\
\delta_2(\langle\langle a,e,o,a\rangle,\langle a,e,e,a\rangle\rangle,\sigma) & = & \langle a,o,e,a\rangle & \delta_2(\langle\langle a,o,o,a\rangle,\langle a,o,o,a\rangle\rangle,\sigma) & = & \langle a,o,o,a\rangle
\end{array}
$$

In a similar way, the set $\Lambda = \{\lambda_k \mid k \geq 0\}$ has to be defined.

$$
\begin{array}{rclclrcl}
\lambda_0(\langle\rangle,\alpha) & = & \eta_1(\alpha) & = & \varepsilon_0\alpha' & \lambda_2(\langle\langle a,e,o,a\rangle,\langle a,e,o,a\rangle\rangle,\sigma) & = & \varepsilon_2\sigma' \\
\lambda_0(\langle\rangle,\beta) & = & \eta_1(\beta) & = & \varepsilon_0\beta' & \lambda_2(\langle\langle a,e,o,a\rangle,\langle a,o,e,a\rangle\rangle,\sigma) & = & \varepsilon_2\sigma' \\
\lambda_1(\langle\langle a,e,e,a\rangle\rangle,\gamma) & = & \eta_3(\gamma) & = & \varepsilon_1\gamma_2' & \lambda_2(\langle\langle a,e,o,a\rangle,\langle a,o,o,a\rangle\rangle,\sigma) & = & \varepsilon_2\sigma' \\
\lambda_1(\langle\langle a,e,o,a\rangle\rangle,\gamma) & = & \eta_{2/3}(\gamma) & (\#) & & \lambda_2(\langle\langle a,o,e,a\rangle,\langle a,e,e,a\rangle\rangle,\sigma) & = & \varepsilon_2\sigma' \\
\lambda_1(\langle\langle a,o,e,a\rangle\rangle,\gamma) & = & \eta_{2/3}(\gamma) & (\#) & & \lambda_2(\langle\langle a,o,e,a\rangle,\langle a,e,o,a\rangle\rangle,\sigma) & = & \varepsilon_2\sigma' \\
\lambda_1(\langle\langle a,o,o,a\rangle\rangle,\gamma) & = & \eta_2(\gamma) & = & \varepsilon_1\gamma_1' & \lambda_2(\langle\langle a,o,e,a\rangle,\langle a,o,e,a\rangle\rangle,\sigma) & = & \varepsilon_2\sigma' \\
\lambda_2(\langle\langle a,e,e,a\rangle,\langle a,e,e,a\rangle\rangle,\sigma) & = & \eta_4(\sigma) & = & \varepsilon_2\sigma' & \lambda_2(\langle\langle a,o,e,a\rangle,\langle a,o,o,a\rangle\rangle,\sigma) & = & \varepsilon_2\sigma' \\
\lambda_2(\langle\langle a,e,e,a\rangle,\langle a,e,o,a\rangle\rangle,\sigma) & = & \eta_4(\sigma) & = & \varepsilon_2\sigma' & \lambda_2(\langle\langle a,o,o,a\rangle,\langle a,e,e,a\rangle\rangle,\sigma) & = & \varepsilon_2\sigma' \\
\lambda_2(\langle\langle a,e,e,a\rangle,\langle a,o,e,a\rangle\rangle,\sigma) & = & \eta_4(\sigma) & = & \varepsilon_2\sigma' & \lambda_2(\langle\langle a,o,o,a\rangle,\langle a,e,o,a\rangle\rangle,\sigma) & = & \varepsilon_2\sigma' \\
\lambda_2(\langle\langle a,e,e,a\rangle,\langle a,o,o,a\rangle\rangle,\sigma) & = & \eta_4(\sigma) & = & \varepsilon_2\sigma' & \lambda_2(\langle\langle a,o,o,a\rangle,\langle a,o,e,a\rangle\rangle,\sigma) & = & \varepsilon_2\sigma' \\
\lambda_2(\langle\langle a,e,o,a\rangle,\langle a,e,e,a\rangle\rangle,\sigma) & = & \eta_4(\sigma) & = & \varepsilon_2\sigma' & \lambda_2(\langle\langle a,o,o,a\rangle,\langle a,o,o,a\rangle\rangle,\sigma) & = & \varepsilon_2\sigma'
\end{array}
$$

In the two lines that are marked with (#), it is irrelevant whether η_2 or η_3 is chosen, because neither 2 nor 3 fulfills the condition for i in (23) on page 70. Let me explain the other two cases: $\lambda_1(\langle\langle a,e,e,a\rangle\rangle,\gamma) = \eta_3(\gamma)$ because $e \in F_3$ and $e \notin F_2$, and $\lambda_1(\langle\langle a,o,o,a\rangle\rangle,\gamma) = \eta_2(\gamma)$ because $o \in F_2$ and $o \notin F_3$.
It is now an interesting question, why the cases $\lambda_1(\langle\langle a,e,o,a\rangle\rangle,\gamma)$ and $\lambda_1(\langle\langle a,o,e,a\rangle\rangle,\gamma)$ are irrelevant. The answer is: because the states $\langle a,e,o,a\rangle$ and $\langle a,o,e,a\rangle$ are inaccessible.
The encouraged reader is invited to check this, and also that the transduction $T^{\mathfrak{R}}$ realized by this sequential relabeling is indeed identical to T as described above.

9.16 Remark It should be obvious that Theorem 9.13 is not a generalization of the characterization for gsm's because the output of an n-seqrel consists of only one symbol in each step, whereas a gsm can have an arbitrary large (but finite) output.

9.17 In Definition 9.12, I used the set $\Sigma^{\odot}$ to define the output of an n-seqrel. Next, I will define for a given ranked alphabet the set $\Sigma^{\blacktriangle}$, which will be used in the definition of sequential forest transducer. Since the formal defintition is quite technical, let me first describe the set $\Sigma^{\blacktriangle}$ informally. Similar to the set $\Sigma^{\odot}$, the set $\Sigma^{\blacktriangle}$ is an $\mathbb{N}$-indexed set with $\Sigma^{\blacktriangle}_k \subseteq \Sigma^{\circledast}[k,1]$ for every $k \in \mathbb{N}$. The set $\Sigma^{\blacktriangle}_k$ contains all those $(k,1)$-forests which contain for every $i = 1, \ldots, n$ the symbol π^n_i exactly once, and for every $j < i$, the symbol π^n_j occurs before π^n_i. If a forest $t \in \Sigma^{\blacktriangle}_k$ is drawn as a box, then each of the k holes of the box is connected to exactly one open position of the tree in the box, and the connections are not allowed to cross.

9.18 Let me now give the formal definition of the set $\Sigma^{\blacktriangle}$. For every $l, m, k \in \mathbb{N}$ with $0 \leq l \leq m \leq k$ and every ranked alphabet Σ, define the set $S(\Sigma, l, m, k) \subseteq \Sigma^{\circledast}[k,1]$ inductively as follows.

- If $l = m$, then for every $\sigma \in \Sigma_0$, let $\langle\rangle_k\sigma \in S(\Sigma, l, m, k)$.
- If $l + 1 = m$, then $\pi^k_m \in S(\Sigma, l, m, k)$.
- For every $k' \geq 0$ and $\sigma \in \Sigma_{k'}$, if there are $t_1, \ldots t_{k'} \in \Sigma^{\circledast}[k,1]$ and $l_0, \ldots, l_{k'} \in \mathbb{N}$ such that
  - $l = l_0 \leq l_1 \leq \ldots \leq l_{k'} = m$ and
  - $\forall i \in \{1, \ldots, k'\}\ .\ t_i \in S(\Sigma, l_{i-1}, l_i, k)$,

  then $\langle t_1, \ldots t_{k'}\rangle\sigma \in S(\Sigma, l, m, k)$.
- There are no other elements in $S(\Sigma, l, m, k)$.

For every ranked alphabet Σ and $k \in \mathbb{N}$, let $\Sigma^{\blacktriangle}_k = S(\Sigma, 0, k, k)$.

9.19 Example Consider again $\Sigma = \Sigma_0 \cup \Sigma_2$ with $\Sigma_0 = \{\alpha\}$ and $\Sigma_2 = \{\sigma\}$. Then, for example,

$$\begin{array}{rcl}
\langle\rangle_0\alpha \in \Sigma^{\blacktriangle}_0 & \text{but} & \langle\rangle_2\alpha \notin \Sigma^{\blacktriangle}_0, \\
\langle\pi^2_1, \pi^2_2\rangle\sigma \in \Sigma^{\blacktriangle}_2 & \text{but} & \langle\pi^2_2, \pi^2_2\rangle\sigma \notin \Sigma^{\blacktriangle}_2, \\
\langle\langle\alpha, \pi^3_1\rangle\sigma, \langle\pi^3_2, \pi^3_3\rangle\sigma\rangle\sigma \in \Sigma^{\blacktriangle}_3 & \text{but} & \langle\langle\alpha, \pi^3_2\rangle\sigma, \langle\pi^3_3, \alpha\rangle\sigma\rangle\sigma \notin \Sigma^{\blacktriangle}_3, \\
\langle\langle\pi^2_1, \pi^2_2\rangle\sigma, \langle\rangle_2\alpha\rangle\sigma \in \Sigma^{\blacktriangle}_2 & \text{but} & \langle\langle\pi^2_2, \pi^2_1\rangle\sigma, \langle\rangle_2\alpha\rangle\sigma \notin \Sigma^{\blacktriangle}_2.
\end{array}$$

9.20 The definition of the set $\Sigma^{\blacktriangle}_k \subseteq \Sigma^{\circledast}[k,1]$ is motivated by the properties stated in the following lemma, which will be needed in the proof of Theorem 9.26.

9.21 Lemma *Let Σ be a ranked alphabet and $n, k \in \mathbb{N}$. Then*

(i) For every $L \subseteq \Sigma^{\circledast}[n,k]$ and $\nu \in \Sigma^{\circledast}[k,1]$, if $L = L_1 \times \ldots \times L_k$ with $L_1, \ldots, L_k \subseteq \Sigma^{\circledast}[n,1]$ and $\nu \in \Sigma^{\blacktriangle}_k$, then

$$L \cdot \nu = L; \nu.$$

(ii) For every $t \in \Sigma^{\circledast}[n,k]$ and $\nu \in \Sigma^{\blacktriangle}_k$,

$$\mathit{size}(t; \nu) = \mathit{size}(t) + \mathit{size}(\nu).$$

Proof: The first property follows from the fact that for every $i = 1, \ldots, k$, the according π_i^k occurs at most once in ν. Then, we have $L \cdot \nu = \{\langle t_1, \ldots, t_k\rangle; \nu \mid t_1 \in L_1, \ldots, t_k \in L_k\} = L; \nu$. The second property simply restates the fact that each π_i^k occurs exactly once in ν. □

9.22 Definition Let $\mathfrak{S} = \langle Q, \Sigma, \Omega, \vec{q}_I, \Delta, \Lambda\rangle$ be an n-dbuft. Then $\mathfrak{S}$ is called a *sequential n forest transducer* (n-*seqft*) if for every $k \geq 0$, every $\sigma \in \Sigma_k$, and every $q_1, \ldots, q_k \in Q$,

$$\lambda_k(\langle q_1, \ldots, q_k\rangle, \sigma) \in \Omega_k^{\blacktriangle}.$$

9.23 Example The 2-dbuft from Example 9.2 is an 2-seqft.

9.24 Remark Following 9.8, a sequential forest transducer is (in terms of [Eng75]) a deterministic, linear, and non-deleting bottom-up tree transducer, where the set of final states F is equal to the set of states Q, and where the variables on the right-hand side of each rule must not permute.

9.25 Remark For every $n \in \mathbb{N}$, an n-seqrel is also an n-seqft, because $\Sigma_k^{\odot} \subseteq \Sigma_k^{\blacktriangle}$ for every ranked alphabet Σ and every $k \geq 0$. Therefore, Theorem 9.13 is a special case of the following theorem.

9.26 Theorem *Let Σ and Ω be ranked alphabets, and let n be a natural number. Let $T = \{\tau_m \mid m \geq 0\}$ be a family of functions $\tau_m : \Sigma^{\circledast}[n, m] \to \Omega^{\circledast}[n, m]$. Then T is a transduction realized by an n-seqft if and only if T satisfies the following five conditions.*

(i) For every $i = 1, \ldots, n$,
$$\tau_1(\pi_i^n) = \pi_i^n.$$

(ii) For every $k \in \mathbb{N}$, every $t \in \Sigma^{\circledast}[n, k]$, and every $\sigma \in \Sigma_k$, there is a $\nu \in \Omega_k^{\blacktriangle}$ such that
$$\tau_1(t\sigma) = \tau_k(t); \nu.$$

(iii) There is a $z \in \mathbb{N}$ such that for every $k \in \mathbb{N}$, every $t \in \Sigma^{\circledast}[n, k]$, and every $\sigma \in \Sigma_k$,
$$size(\tau_1(t\sigma)) - size(\tau_k(t)) < z.$$

(iv) For every $m \in \mathbb{N}$, if $L \in Rat(\Omega, n, m)$, then $\tau_m^{-1}(L) \in Rat(\Sigma, n, m)$.

(v) For every $m \geq 0$ and $t_1, \ldots, t_m \in \Sigma^{\circledast}[n, 1]$,
$$\tau_m(\langle t_1, \ldots, t_m\rangle) = \langle \tau_1(t_1), \ldots, \tau_1(t_m)\rangle.$$

Proof: The first direction (the "if"-part) is very similar to the proof of Theorem 9.13. Assume, there is an n-seqft $\mathfrak{S} = \langle Q, \Sigma, \Omega, \vec{q}_I, \Delta, \Lambda\rangle$ with $T^{\mathfrak{S}} = T$. It is easy to see that $T^{\mathfrak{S}}$ fulfills Conditions (i),(ii),(iv), and (v) of the theorem (see proof of Theorem 9.13). For (iii), let

$$z = max\{size(\nu) \mid \exists k \geq 0 \;.\; \exists q_1, \ldots, q_k \in Q \;.\; \exists \sigma \in \Sigma_k \;.\; \nu = \lambda_k(\langle q_1, \ldots q_k\rangle, \sigma)\} + 1.$$

Then z certainly satisfies Condition (iii).

The proof of the other direction (the "only if"-part) also follows the proof of Theorem 9.13. Assume that $T = \{\tau_m \mid m \in \mathbb{N}\}$ is a family of functions $\tau_m : \Sigma^{\circledast}[n,m] \to \Omega^{\circledast}[n,m]$ that satisfy Conditions (i) to (v) of Theorem 9.26. Then Condition (ii) induces for every $k \geq 0$ a function

$$f_k : \Sigma^{\circledast}[n,k] \times \Sigma_k \to \Omega_k^{\blacktriangle},$$

which can be curried to

$$\hat{f}_k : \Sigma^{\circledast}[n,k] \to [\Sigma_k \to \Omega_k^{\blacktriangle}].$$

Now define for every $k \in \mathbb{N}$ the set

$$H_k = im(\hat{f}_k) = \{\eta \in [\Sigma_k \to \Omega_k^{\blacktriangle}] \mid \exists t \in \Sigma^{\circledast}[n,k] \ . \ \hat{f}_k(t) = \eta\}$$

to be the image of the function $\hat{f}_k$. Then, for every $k \geq 0$, the set H_k is finite because Condition (iii) of Theorem 9.26 restricts the image of every function $\eta \in H_k$ to a finite set. Now, for every $k \geq 0$ and every $\eta \in H_k$, the language $L_k(\eta) \subseteq \Sigma^{\circledast}[n,k]$ is defined in the same way as in the proof of Theorem 9.13.

$$L_k(\eta) = \{t \in \Sigma^{\circledast}[n,k] \mid \hat{f}_k(t) = \eta\}.$$

Again, for every $k \geq 0$ there is only a finite number of such languages because H_k is finite. Furthermore, it holds the following claim.

Claim 1: For every $k \geq 0$, The set $\Sigma^{\circledast}[n,k]$ is the disjoint union of all $L_k(\eta)$ with $\eta \in H_k$, i.e.

$$\Sigma^{\circledast}[n,k] = \bigcup_{\eta \in H_k} L_k(\eta) \ \text{ and for every } \eta_1, \eta_2 \in H_k \text{ with } \eta_1 \neq \eta_2,\ L_k(\eta_1) \cap L_k(\eta_2) = \emptyset.$$

Proof of Claim 1: As in the proof of Theorem 9.13.

The difficulty of this proof occurs in the next step, where we have to show that for every $k \geq 0$ and $\eta \in H_k$, the language $L_k(\eta) \subseteq \Sigma^{\circledast}[n,k]$ is semi-recognizable. For every $k \geq 0$, $\sigma \in \Sigma_k$, and $\nu \in \Omega_k^{\blacktriangle}$, define the language

$$N_k(\sigma,\nu) = \{t \in \Sigma^{\circledast}[n,k] \mid \tau_1(t\sigma) = \tau_k(t);\nu\}$$

Claim 2: $N_k(\sigma,\nu) \in SRec(\Sigma,n,k)$.

Proof of Claim 2: Clearly, now the situation is not as trivial as in the proof of Theorem 9.13, but the main idea will be the same, i.e. I will define a semi-recognizable language $N'_k(\sigma,\nu)$ and show that it is equal to $N_k(\sigma,\nu)$.
First, define for every $\zeta < z$ (the z from Condition (iii)) the set

$$W_1(\zeta) \quad = \quad \{t \in \Omega^{\circledast}[n,1] \mid \exists p \geq 1 \ . \ size(t) = p \cdot z + \zeta\}. \tag{24}$$

Let me assert that $W_1(\zeta) \in Rat(\Omega^{\circledast},n,1)$ and postpone the proof of this statement (see Claim 3).
Further, let Let $V = \{t \in \Omega^{\circledast}[n,1] \mid size(t) < z\}$. Then V is finite. Let $x = |V|$, i.e. $V = \{v_1,\ldots,v_x\}$. Now define for every $\xi = 1,\ldots,x$ the singleton

$$W_2(\xi) = \{v_\xi\}.$$

Let $M = \{(1,\zeta) \mid \zeta < z\} \cup \{(2,\xi) \mid 1 \leq \xi \leq x\}$, which is a finite set. For every $\mu \in M$ define

$$W(\mu) = \begin{cases} W_1(\zeta) & \text{if } \mu = (1,\zeta) \\ W_2(\xi) & \text{if } \mu = (2,\xi) \end{cases}$$

Now define the language

$$N'_k(\sigma,\nu) = \bigcup_{\langle\mu_1,\ldots,\mu_k\rangle \in M^k} d_\sigma(\tau_1^{-1}((W(\mu_1) \times \ldots \times W(\mu_k));\nu)) \cap \tau_k^{-1}(W(\mu_1) \times \ldots \times W(\mu_k)).$$

It is clear (by Lemma 9.21.(i)) that $N'_k(\sigma,\nu)$ is semi-recognizable under the condition that for every $\zeta < z$, the set $W(\zeta)$ is rational.

Let me now show that $N_k(\sigma,\nu) = N'_k(\sigma,\nu)$. First, let $t \in N_k(\sigma,\nu)$, in other words, $\tau_1(t\sigma) = \tau_k(t);\nu$. Since

$$\Sigma^{\circledast}[n,1] = \bigcup_{\mu \in M} W(\mu),$$

there are $\mu_1,\ldots,\mu_k \in M$ such that $\tau_k(t) \in W(\mu_1)\times\ldots\times W(\mu_k)$. Hence, $t \in \tau_k^{-1}(W(\mu_1)\times\ldots\times W(\mu_k))$. Furthermore, $\tau_1(t\sigma) = \tau_k(t);\nu \in (W(\mu_1)\times\ldots\times W(\mu_k));\nu$. Hence, $t \in d_\sigma(\tau_1^{-1}((W(\mu_1)\times\ldots\times W(\mu_k));\nu)$. Therefore, $t \in N'_k(\sigma,\nu)$.

For the other direction, assume $t \in N'_k(\sigma,\nu)$ and let $s = \tau_k(t) \in \Omega^{\circledast}[n,k]$. Then there are $\mu_1,\ldots,\mu_k \in M$ such that

$$\begin{aligned}
\tau_k(t) &\in W(\mu_1)\times\ldots\times W(\mu_k), \quad \text{and} \\
\tau_1(t\sigma) &\in (W(\mu_1)\times\ldots\times W(\mu_k));\nu.
\end{aligned}$$

In other words, we have $s \in W(\mu_1)\times\ldots\times W(\mu_k)$, and there is an $s' \in W(\mu_1)\times\ldots\times W(\mu_k)$ such that $\tau_1(t\sigma) = s';\nu$. Let $s = \langle s_1,\ldots,s_k\rangle$ and $s' = \langle s'_1,\ldots,s'_k\rangle$. This means, we have $s_i, s'_i \in W(\mu_i)$ for every $i = 1,\ldots,k$.

Now, let $K_1, K_2 \subseteq \{1,\ldots,k\}$ be defined as follows.

$$\begin{aligned}
K_1 &= \{i \mid 1 \le i \le k \ \wedge\ \exists\zeta < z \,.\, \mu_i = (1,\zeta)\} \\
K_2 &= \{j \mid 1 \le j \le k \ \wedge\ \exists\xi \le x \,.\, \mu_j = (2,\xi)\}
\end{aligned}$$

Obviously, $K_1 \cup K_2 = \{1,\ldots,k\}$ and $K_1 \cap K_2 = \emptyset$. Further, for every $i \in K_1$ define ζ_i such that $\mu_i = (1,\zeta_i)$ and for every $j \in K_2$ define ξ_j such that $\mu_j = (2,\xi_j)$.
By (24), we have for every $i \in K_1$ two numbers $p_i, q_i \ge 1$ such that $size(s_i) = p_i z + \zeta_i$ and $size(s'_i) = q_i z + \zeta_i$. By Definition 4.43 and Lemma 9.21.(ii),

$$size(\tau_k(t)) = size(s) = size(s_1)+\ldots+size(s_k) = \sum_{i\in K_1}(p_i z+\zeta_i) + \sum_{j\in K_2} size(s_j), \text{ and} \tag{25}$$

$$size(\tau_1(t\sigma)) = size(s'_1)+\ldots+size(s'_k)+size(\nu) = \sum_{i\in K_1}(q_i z+\zeta_i) + \sum_{j\in K_2} size(s'_j) + size(\nu). \tag{26}$$

Furthermore, by (ii), there is a $\nu' \in \Omega_k^{\blacktriangle}$ with $\tau_1(t\sigma) = \tau_k(t);\nu' = s;\nu'$. Hence, by (25) and Lemma 9.21(ii),

$$size(\tau_1(t\sigma)) = size(s_1)+\ldots+size(s_k)+size(\nu') = \sum_{i\in K_1}(p_i z+\zeta_i) + \sum_{j\in K_2} size(s_j) + size(\nu'). \tag{27}$$

Since for every $j \in K_2$ the set $W(\mu_j) = W_2(\xi_j)$ is a singleton, we have $s_j = s'_j$. Hence, from (26) and (27), we obtain

$$\left(\sum_{i\in K_1} q_i\right) z + size(\nu) = \left(\sum_{i\in K_1} p_i\right) z + size(\nu').$$

By Condition (iii), we have $size(\nu) < z$ and $size(\nu') < z$. Hence,

$$\sum_{i\in K_1} q_i = \sum_{i\in K_1} p_i \qquad \text{and} \qquad size(\nu) = size(\nu').$$

We can even conclude that for every $i \in K_1$, $q_i = p_i$. To see this suppose that there is an $i \in K_1$ such that $q_i \ne p_i$. Then $size(s'_i) \ge size(s_i) + z$ or $size(s_i) \ge size(s'_i) + z$. Both cases are impossible because

$size(\nu) < z$ and $s;\nu' = s';\nu$ and because ν and ν' must not change the order of the $\pi_1^k, \ldots, \pi_k^k$.
Therefore, for every $i \in K_1$, we have $size(s_i) = size(s'_i)$. It follows that for every $i \in K_1$, we have $s_i = s'_i$ because $size(s_i) = size(s'_i) > size(\nu) = size(\nu')$. And since we already know that $s_j = s'_j$ for every $j \in K_2$, we get $s' = s = \tau_k(t)$. Note that it does not follow that $\nu = \nu'$. However, since $s' = \tau_k(t)$ and $\tau_1(t\sigma) = s';\nu$, we obtain $t \in N_k(\sigma, \nu)$, which completes the proof of Claim 2.
It remains to show

Claim 3: For every $\zeta < z$, $W_1(\zeta) \in Rat(\Omega^{\circledast}, n, 1)$.

Proof of Claim 3: It is easy to see that the set $W' = \{t \in \Omega^{\circledast}[n,1] \mid \exists p \in \mathbb{N} \,.\, size(t) = p \cdot z + \zeta\}$ is a recognizable language. The according (deterministic) automaton has z states $\{0, \ldots, z-1\}$ with $\langle 0, \ldots, 0 \rangle$ as initial state tuple, ζ as final state, and for every $k \geq 0$, the state transition function δ_k realizes the addition of k numbers modulo z. Precisely, for every $q_1, \ldots, q_k < z$ and $\sigma \in \Sigma_k$,

$$\delta_k(\langle q_1, \ldots, q_k \rangle, \sigma) = q_1 + \ldots + q_k + 1 \ (mod\ z).$$

Next, the set $W'' = \{t \in \Omega^{\circledast}[n,1] \mid size(t) < z\}$ is recognizable because it is finite. Obviously, $W_1(\zeta) = W' \setminus W''$, which is recognizable (by Proposition 6.1). By Theorem 7.9, we get $W_1(\zeta) \in Rat(\Omega^{\circledast}, n, 1)$, which completes the proof of Claim 3.

Let me now come back to the languages $L_k(\eta)$ as defined above. As in the proof of Theorem 9.13, we obtain for every $k \geq 0$ and $\eta \in H_k$,

$$L_k(\eta) = \bigcap_{\sigma \in \Sigma_k} N_k(\sigma, \eta(\sigma)).$$

Therefore, $L_k(\eta) \in SRec(\Sigma, n, k)$.
The construction of the sequential forest transducer that realizes T is exactly the same as the construction in the proof of Theorem 9.13 (page 69). For that reason, I will not repeat it here. □

9.27 At this point, the question arises, whether the rather strong restriction of not allowing any permutation in the output is really necessary. It turns out that this question has a positive answer, which will be illustrated by the following example.

9.28 Example Let $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$ with $\Sigma_0 = \{\alpha\}$, $\Sigma_1 = \{\gamma\}$, and $\Sigma_2 = \{\sigma\}$, and let $n = 0$. Now, let $T = \{\tau_m \mid m \geq 0\}$ be a family of functions $\tau_m : \Sigma^{\circledast}[0,m] \to \Sigma^{\circledast}[0,m]$ such that Condition (v) of Theorem 9.26 is satisfied. Furthermore,

- let $\tau_1(\langle\rangle_0 \alpha) = \langle\rangle_0 \alpha$,
- for every $t \in \Sigma^{\circledast}[0,1]$, let $\tau_1(t\gamma) = \tau_1(t)\gamma$, and
- for every $t = \langle t_1, t_2 \rangle \in \Sigma^{\circledast}[0,2]$, let

$$\tau_1(t\sigma) = \begin{cases} \tau_2(t); \langle \pi_1^2, \pi_2^2 \rangle \sigma = \langle \tau_1(t_1), \tau_1(t_2) \rangle \sigma & \text{if } size(t_1) \geq size(t_2) \\ \tau_2(t); \langle \pi_2^2, \pi_1^2 \rangle \sigma = \langle \tau_1(t_2), \tau_1(t_1) \rangle \sigma & \text{if } size(t_1) < size(t_2) \end{cases}.$$

The transduction $T = \{\tau_m \mid m \geq 0\}$ cannot be realized by any dbuft because, informally speaking, a dbuft has only a finite memory and is therefore not able to compare the size of two arbitrary trees. However, T clearly satisfies Conditions (i), (iii), and (v) of Theorem 9.26. It almost fulfills Condition (ii), except that $\langle \pi_2^2, \pi_1^2 \rangle \sigma \notin \Sigma_2^{\blacktriangle}$. Let me now show that T does satisfy Condition (iv). By Lemma 7.5, it suffices to show the case $m = 1$. So, let $L \in Rat(\Sigma, n, 1)$. Then, by Theorem 7.9, $L \in Rec(\Sigma, n, 1)$, i.e. there is a $(0,1)$-fsfa $\mathfrak{A} = \langle Q, \Sigma, I, F, E \rangle$ where $I = \langle\rangle$, $E \subseteq Q$ and $E = \{E_k \mid k \geq 0\}$ such that $L(\mathfrak{A}) = L$. Let me now construct a $(0,1)$-fsfa

$\mathfrak{A}'$ with $L(\mathfrak{A}') = \tau_1^{-1}(L)$. Let $\mathfrak{A}' = \langle Q, \Sigma, I, F, E' \rangle$ with $E' = \{E'_k \mid k \geq 0\}$, where $E'_0 = E_0$, $E'_1 = E_1$, and

$$E'_2 = E_2 \; \cup \{(\langle q_2, q_1 \rangle, \sigma, q) \mid (\langle q_1, q_2 \rangle, \sigma, q) \in E_2\}.$$

Then a forest $t \in \Sigma^{\circledast}[0,1]$ is contained in $L(\mathfrak{A}')$ iff it can be obtained from a forest $t' \in L$ by exchanging the two subtrees of some σ's that occur in t'. This is exactly the language, which is described by $\tau_1^{-1}(L)$. Hence, $\tau_1^{-1}(L) \in \mathit{Rec}(\Sigma, n, 1)$. Therefore, by Theorem 7.9, $\tau_1^{-1}(L) \in \mathit{Rat}(\Sigma, n, 1)$. Hence, T satisfies Condition (iv) of Theorem 9.26.
This example shows that if we drop the requirement of not having any permutation of the occurances of $\pi_1^k, \ldots, \pi_k^k$ in the forests in $\Sigma_k^{\blacktriangle}$, then we cannot hope to obtain a generalization of Theorem 9.26.

9.29 Remark Let me now show that the result by Ginsburg and Rose is indeed a special case of Theorem 9.26. Assume two ranked alphabets Σ and Ω that contain only symbols of rank one, i.e. $\Sigma = \Sigma_1$ and $\Omega = \Omega_1$. Now consider an 1-seqft $\mathfrak{S} = \langle Q, \Sigma, \Omega, \vec{q}_I, \Delta, \Lambda \rangle$. Since $\Sigma = \Sigma_1$, the only interesting function in Δ is $\delta_1 : Q \times \Sigma \to Q$, and the only interesting function in Λ is $\lambda_1 : Q \times \Sigma \to \Omega_1^{\blacktriangle}$. Since $\Omega = \Omega_1$, we have $\Omega_1^{\blacktriangle} = \Omega^{\circledast}[1,1] = \Omega^*$. Hence, $\mathfrak{S}$ is a gsm and the transduction $T^{\mathfrak{S}} = \{\tau_m^{\mathfrak{S}} \mid m \in \mathbb{N}\}$ is reduced to $\tau_1^{\mathfrak{S}} : \Sigma^{\circledast}[1,1] \to \Omega^{\circledast}[1,1]$. Under this condition, Theorem 9.26 reduces to the well-known characterization of gsm's.

10 Conclusion

I started to think about finite-state automata on trees because I was asked to find a generalization of the characterization of gsm's for some class of tree transducers. It turned out that a naive approach to generalizing the construction in [GR66] to trees must fail. There are three reasons:

- The tree concatenation has different properties than the word concatenation.
- It is not possible to define the right quotient of a tree language in the same way as the right quotient of a string language is defined.
- The rational expressions for tree languages cannot be treated in the same way as the rational expressions for string languages.

The concept of forests could solve those problems. However, in order to find the generalization of the characterization for gsm's, it became necessary to define the notions of recognizability and rationality for forest languages.

I extended the notion of finite-state tree automaton to that of finite-state forest automaton by adding initial states and extending the final states to final state tuples. This led to the definition of recognizable forest languages. The rational forest languages are defined by means of rational expressions in which the four operations of union $\cup$, cartesian product $\times$, forest language concatenation $\cdot$, and forest language concatenation closure * can occur. There is no need to extend the alphabet and to use many different concatenation operations. The first contribution of this thesis is showing that that the recognizable and the rational forest languages are equal.

The use of forests instead of trees makes it necessary to extend the notion of tree transducer to that of forest transducer. Only the fact that the recognizable and the rational forest languages are equal justifies the attempt to find a "machine independent" characterization for a certain class of forest transducers.

The straightforward extension of the definition of generalized sequential machines (gsm) led to the definition of deterministic bottom-up forest transducers (dbuft) (see Definition 9.1), for which the extension of the characterization does not hold. For this reason, I introduced the class of sequential forest transducers (seqft) (see Definition 9.22), which is a subclass of the class of dbuft's. The second contribution in this thesis is a "machine independent" characterization for seqft's, which is indeed a generalization of the characterization for gsm's. Furthermore, Example 9.28 shows that there is no hope of extending that characterization to a broader class of forest transducers.

However, in the field of tree transducers, we can find enough space for further investigation. Every tree transducer can be extended to a forest transducer by a straightforward construction. This means that there is a chance of obtaining other "machine independent" characterizations for some tree transducers by extending them to forest transducers, in the same way as it has been done in the case of sequential forest transducers in this work. I would consider it to be fruitful to investigate rational forest transductions as a generalization of the rational transductions which are investigated in [Ber79, ch.III].

Example 4.10 shows that function clones (cf. [PK79]) can be seen as polyoids. This means that is is possible with the approach in this thesis to define the recognizable and rational sets in function clones. I consider it to be worth while to investigate where there are other function clones, except for the free one, in which the recognizable and the rational sets are equal.

Other possibilities of further investigation are in the field of automata theory. Since the recognizable forest languages are isomorphic to cartesian products of recognizable tree languages, it should be possible to extend the whole theory of recognizable tree languages, (e.g., pumping lemma, regular grammars, Myhill-Nerode theorem, monadic second order logic) to recognizable forest languages.

Acknowledgements

This thesis has been written as part of my course of studies in the International Masters Programme in Computational Logic. I want to thank all the people involved in keeping this programme running, because without them this thesis would not have been written. In particular, I would like to name Steffen Hölldobler who organized the programme and Maja von Wedelstedt, the secretary of the programme.

I gratefully thank my supervisor Heiko Vogler who taught me how to write a paper and who found the time to struggle through early versions of this thesis. Without his comments and suggestions, I would not have been able to get through.

I also want to thank Manfred Droste who awoke my interest for the fascinating field of automata theory and who gave useful comments on an early version of this thesis. Horst Reichel gave me helpful insights into universal algebra and category theory. Alessio Guglielmi listened to me and made good comments and suggestions that helped me to improve the readability of the thesis. Jan Rothe made good remarks on the introduction and parts of an early version.

Last, but not least, I warmly thank my parents for supporting me all the time and for believing in me when I didn't.

References

[Ber79] Jean Berstel. *Transductions and Context-Free Languages*. Number 38 in Leitfäden der angewandten Mathematik und Mechanik LAMM. B.G. Teubner Stuttgart, 1979.

[BW99] Micheal Barr and Charles Wells. *Category Theory for Computing Science*. Les Publications CRM, Montréal, third edition, 1999.

[DG99] M. Droste and P. Gastin. The Kleene–Schützenberger theorem for formal power series in partially commuting variables. *Information and Computation*, 153:47–80, 1999.

[Eil76] S. Eilenberg. *Automata, Languages And Machines*, volume B. Academic Press, New York, San Francisco, London, 1976.

[Eng75] Joost Engelfriet. Bottom-up and top-down tree transformations—a comparison. *Mathematical Systems Theory*, 9(3):198–231, 1975.

[Eng77] Joost Engelfriet. Top-down tree transducers with regular look-ahead. *Mathematical Systems Theory*, 10:289–303, 1977.

[FV98] Zoltán Fülöp and Heiko Vogler. *Syntax-Directed Semantics: Formal Models Based on Tree Transducers*. Springer-Verlag Berlin Heidelberg, 1998.

[GR66] S. Ginsburg and G. F. Rose. A characterisation of machine mappings. *Can. J. of Math.*, 18:381–388, 1966.

[GS97] Ferenc Gécseg and Magnus Steinby. Tree Languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages, Beyond Words*, volume 3, chapter 1, pages 1–68. Springer-Verlag Berlin Heidelberg, 1997.

[Kle56] S. C. Kleene. Representation of events in nerve nets and finite automata. In C. E. Shannon and J. McCarthy, editors, *Automata Studies*, pages 3–40. Princeton, N.J., 1956.

[Kun80] Kenneth Kunen. *Set Theory, An Introduction to Independence Proofs*. Elsevier Science Publishers B.V., 1980.

[Law63] F. W. Lawvere. Functorial semantics of algebraic theories. In *Proc. National Academy of Science*, 50, pages 869–872, 1963.

[Och85] E. Ochmański. Regular behaviour of concurrent systems. *Bull. Europ. Assoc. Theoret. Comput. Sci. (EATCS)*, 27:56–67, 1985.

[Per90] D. Perrin. Finite Automata. In J. v. Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, Formal Models and Semantics, chapter 1, pages 1–57. Elsevier Science Publishers, B.V., 1990.

[PK79] R. Pöschel and L. A. Kalužnin. *Funktionen- und Relationenalgebren, Ein Kapitel der Diskreten Mathematik*. Deutscher Verlag der Wissenschaften, Berlin, 1979.

[Sch61] M. P. Schützenberger. On the definition of a family of automata. *Inform. And Control*, 4:245–270, 1961.

[TW68] J. W. Thatcher and J. B. Wright. Generalized finite automata theory with an application to a decision problem of second-order logic. *Math. Systems Theory*, 2:57–81, 1968.

[Wec92] Wolfgang Wechler. *Universal Algebra for Computer Scientists.* Springer Verlag Berlin Heidelberg, 1992.

Erklärung

Hiermit erkläre ich, daß ich die vorliegende Arbeit selbständig verfaßt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Dresden, den 6. April 2000

Lutz Straßburger