December 17, 2008 — Final version for proceedings of LATA'09

A Kleene Theorem for Forest Languages

Lutz Straßburger

INRIA Saclay – Île-de-France — Équipe-projet Parsifal
École Polytechnique — LIX — Rue de Saclay — 91128 Palaiseau Cedex — France
http://www.lix.polytechnique.fr/~lutz

Abstract. This paper proposes an alternative approach to the standard notion of rational (or regular) expression for tree languages. The main difference is that in the new notion we have only one concatenation operation and only one star-operation, instead of many different ones. This is achieved by considering forests instead of trees over a ranked alphabet, or, algebraicly speaking, by considering cartesian categories instead of term-algebras. The main result is that in the free cartesian category the rational languages and the recognizable languages coincide. For the construction of the rational expression for a recognizable language it is not necessary to extend the alphabet. We only use operations that can be defined with the algebraic structure provided by cartesian categories.

1 Introduction

Kleene's theorem on the coincidence of the rational and the recognizable languages in a free monoid [Kle56] is considered to be one of the cornerstones of theoretical computer science. Although it does not hold in every monoid [Eil76], it has been generalized in several directions, e.g., [Och85,Sch61,DG99]. Thatcher and Wright presented in [TW68] a similar result for tree languages. In recent years, recognizable tree languages have received increased attention because they form a foundation for XML schemas for expressing semi-structured data (e.g., [Nev02,MN07,CDG$^+$07]). In this paper I will only consider ranked trees and tuples of ranked trees, which I call *forests* to distinguish them from *hedges* [Cou89] which are sequences of unranked trees [CDG$^+$07].

The recognizable tree languages are defined by means of *finite-state tree automata* (*fsta*), which are a generalization of ordinary *finite state automata* (*fsa*) over words. While an fsa works on an alphabet A, 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 of all finite trees over Σ. The running example for this paper is the ranked alphabet $\Sigma = \{\sigma, \gamma, \alpha\}$, where σ has rank 2, γ has rank 1, and α has rank 0. Let us define an fsta $\mathfrak{A}$ with two states p and f, where f is a final state. Observe that (contrary to fsa on words) there are no initial states. Usually the behaviour of an fsta is represented as set of rules of a term rewriting system [GS97]. In the example, let the behaviour of $\mathfrak{A}$ be described by the rules

$$\alpha \to p \quad , \qquad \alpha \to f \quad , \qquad \gamma(p) \to p \quad , \qquad \sigma(p, f) \to f \quad .$$

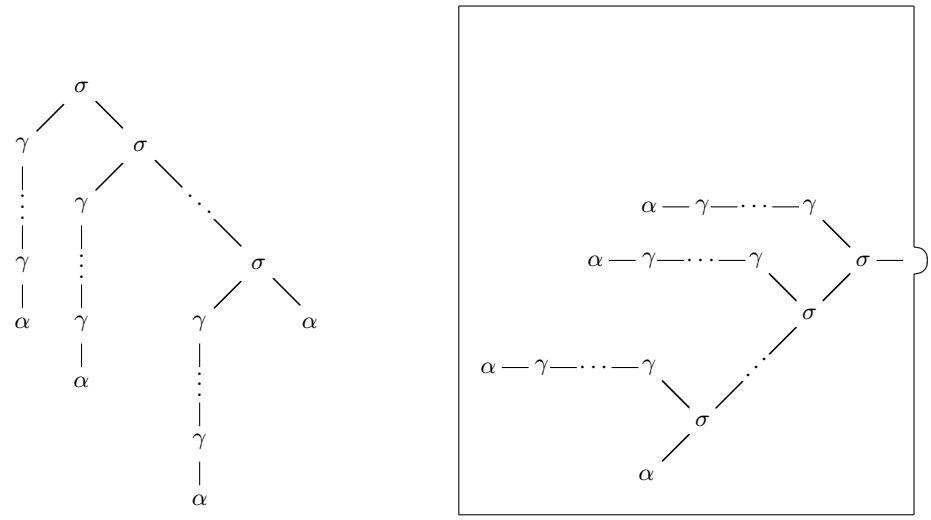

Fig. 1. Left: The form of the trees in language L **Right:** Depicted as forests

The tree language L accepted by this automaton contains the tree α and all trees of the shape $\sigma(\gamma \dots \gamma\alpha, \sigma(\gamma \dots \gamma\alpha, \sigma(\dots, \sigma(\gamma \dots \gamma\alpha, \alpha) \dots)))$, which is more vivid if drawn as a tree as on the left of Fig. 1, where the number of γ's in each chain and the number of σ's can be any natural number (including zero).

In order to find for this recognizable tree language a rational expression in the same sense as for recognizable word languages, 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 approach taken in [TW68] 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 those leaves. In a tree over Σ, the leaves are labeled by the 0-ary symbols of Σ. Hence, for every 0-ary symbol of Σ, there is another concatenation operation. In order to obtain a rational expression for every recognizable tree language, it is necessary to extend Σ by additional 0-ary symbols. The rational expression for the language in our example is

$$L = \{\alpha\} \cup (\{\sigma(p, f)\} \cdot_p \{\gamma(p)\}^{*p} \cdot_p \{\alpha\})^{*f} \cdot_f \{\alpha\} \quad . \tag{1}$$

Observe that there are two additional symbols p and f with rank 0 that are not elements of Σ. This means that 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 (1), are not subsets of T_Σ but subsets of $T_{\Sigma \cup \{p,f\}}$. We see two different concatenation operations, $\cdot_p$ and $\cdot_f$, in (1), because there are two states in the corresponding 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. There are three points that one might find disturbing:

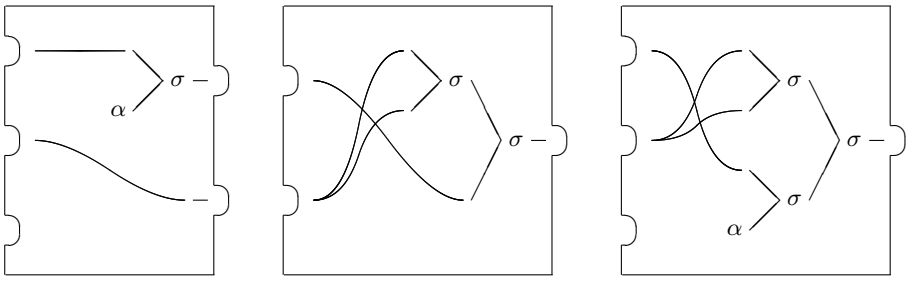

Fig. 2. The three forests s, t and r

(i) The language for formalizing the rational expressions is not predefined by Σ. 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 L. (In rational expressions for word languages we have only one concatenation operation.)

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

(iii) The set T_Σ of all trees over Σ forms the free Σ-algebra generated by the empty set. The tree concatenation operations have no correspondence in the Σ-algebra structure. (The set A^* of all words over A forms the free monoid generated by A, and the word concatenation operation is the multiplication in the monoid.) As a consequence, the notion of rational (tree) language cannot easily be generalized to any Σ-algebra, as it has been done very successfully for rational languages in any monoid [Eil76,Och85,Sch61,DG99].

These three problems can be entirely avoided by a remarkable simple generalization: instead of concatenating trees, we concatenate tuples of trees, i.e., forests. The set of all forests over a ranked alphabet Σ will be denoted by $\Sigma^{\circledast}$. Algebraicly speaking, we generalize the notion of monoid not to a Σ-algebra but to a cartesian category, and $\Sigma^{\circledast}$ is the free cartesian category generated by Σ. The usual free monoid of words can be seen as the special case in which all symbols in Σ have rank 1. One can visualize a forest as a box with holes on one side and plugs on the other. As example, Figure 2 shows three forests (for the same Σ as above). If we call them s, t and r, respectively, then s contains two trees and t and r contain each one tree. The leaves either contain a symbol of Σ with rank 0 or are connected to a hole of the box. The concatenation operation, which is the composition of arrows in the category, becomes now a plugging together of two forests with a matching number of plugs and holes. In our example, s and t can be plugged together and the result is r. With this approach, the many tree concatenation operations are reduced to a single forest concatenation operation.

Note that this category has been around since the work by Lawvere on algebraic theories [Law63]. It has also been extensively studied in the context of

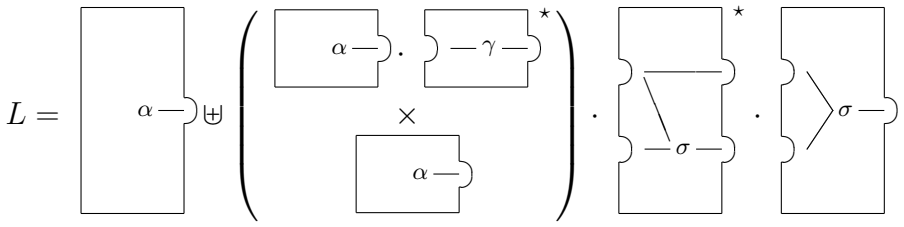

Fig. 3. The rational expression in for the language L

iteration theories [BÉ93], in which the regular forests have been characterized in various ways [Ésik81]. But the structure is different from preclones [ÉW05], which have another concatenation operation (see also Remark 2.2 in [ÉW05]). Our category is also a special case of a magmoid [AD82]. However, it seems that the structure of cartesian category has not yet been used to define the notion of *rational expression* in order to provide a Kleene-theorem (but see [Boj07] for related work on unranked trees).

The notion of recognizability presented in this paper is the same as the one by Thatcher and Wright [TW68] or the one by Ésik and Weil [ÉW05] (modulo the cosmetic difference that we consider forests instead of trees). The novelty is the new notion of rational expression, whose definition is closer to the rational languages in a monoid [Eil76]. The only operations that are allowed are the componentwise union $\uplus$, the cartesian product $\times$, the concatenation $\cdot$, and the concatenation closure *, which is the generalization of the Kleene star * for words. Although it is straightforward to translate the many concatenation and star-operations of [TW68] for trees into the corresponding forest operations presented in this paper, the converse translation is not so immediate. Nonetheless, it follows from [TW68] and our work, that the two notions coincide for tree languages. Thus, this paper provides yet another characterization of the recognizable (or regular) tree languages.

For giving an example, the right-hand side of Fig. 1 shows how the language L can be depicted as forest language. The new rational expression is depicted in Fig. 3. Observe that there is only one concatenation operation and that every subexpression represents a subset of $\Sigma^{\circledast}$. Note that the operations used in a rational expression are available in any cartesian category (a category with a terminal object and all finite products). This means that the notions of recognizability and rationality defined in this paper are available in all cartesian categories, where the two notions do in general not coincide.

Related to this work is the general notion of substitution (e.g., [Ede85]). However, the set of substitutions forms a monoid and the composition of substitutions is the multiplication in that monoid. Imposing the notions of recognizability and rationality on substitutions would mean to use the well-known notions of recognizability and rationality in a monoid [Eil76], whereas the purpose of this paper is to capture the notion of recognizability for trees. For this it is necessary to add to each substitution also the information about which variables occur in the domain and in the codomain; and this in turn leads to the notion of forest.

All the technical details of this paper are available in [Str00].

2 Forests

Forests are tuples of trees (or terms) over a ranked alphabet. In other words, they are the arrows in the free cartesian category generated by the alphabet, and the forest concatenation is the usual term substitution, which is the arrow composition in that category [Law63]. In order to make the paper self-contained, the definitions are given if full (without using category theory).

2.1 Definition A *ranked alphabet* Σ is a finite set of symbols together with a function *rank* : $\Sigma \to \mathbb{N}$, which assigns to every symbol $\sigma \in \Sigma$ its rank. For a given $k \in \mathbb{N}$, let $\Sigma_k = \{\sigma \in \Sigma \mid \mathit{rank}(\sigma) = k\}$.

2.2 Example For the example from the introduction we have $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$ with $\Sigma_0 = \{\alpha\}$, $\Sigma_1 = \{\gamma\}$ and $\Sigma_2 = \{\sigma\}$.

2.3 Definition Let Σ be a ranked alphabet. For every $n, m \geq 0$, the set $\Sigma^{\circledast}[n, m]$ of all (n, m)-*forests over* Σ is the smallest set such that

(i) the set $\Sigma^{\circledast}[n, 0] = \{\langle\rangle_n\}$ is a singleton,
(ii) for every $i \in \{1, \ldots, n\}$ (with $n \geq 1$), there is a forest $\pi^n_i \in \Sigma^{\circledast}[n, 1]$,
(iii) for every $k \geq 0$ and $t \in \Sigma^{\circledast}[n, k]$ and $\sigma \in \Sigma_k$, there is a forest $t\sigma \in \Sigma^{\circledast}[n, 1]$,
(iv) if $m \geq 2$ and $t_1, \ldots, t_m \in \Sigma^{\circledast}[n, 1]$, then $\langle t_1, \ldots, t_m\rangle \in \Sigma^{\circledast}[n, m]$.

The elements of the set $\Sigma^{\circledast} = \bigcup_{n,m\in\mathbb{N}} \Sigma^{\circledast}[n, m]$ are called *forests over* Σ.

2.4 Example Let Σ be as above. Then $s = \langle\langle\pi^3_1, \langle\rangle_3\alpha\rangle\sigma, \pi^3_2\rangle$ is a $(3, 2)$-forest and $t = \langle\langle\pi^2_2, \pi^2_2\rangle\sigma, \pi^2_1\rangle\sigma$ is a $(2, 1)$-forest. Both are depicted in Fig. 2.

2.5 Notation By some abuse of notation, we can see Σ_k as subset of $\Sigma^{\circledast}[k, 1]$, by identifying $\sigma \in \Sigma_k$ with $\langle\pi^k_1, \ldots, \pi^k_k\rangle\sigma \in \Sigma^{\circledast}[k, 1]$.

2.6 Definition For a given ranked alphabet Σ, the *forest concatenation* ; is inductively defined as follows.

(i) For every $n, m \in \mathbb{N}$ and $t \in \Sigma^{\circledast}[n, m]$, let $t; \langle\rangle_m = \langle\rangle_n$,
(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 $i \in \{1, \ldots, m\}$, let $\langle t_1, \ldots, t_m\rangle; \pi^m_i = t_i$,
(iii) for every $n, m, k \in \mathbb{N}$, every $s \in \Sigma^{\circledast}[n, m]$, every $t \in \Sigma^{\circledast}[m, k]$, and every $\sigma \in \Sigma_k$, let $s; (t\sigma) = (s; t)\sigma$,
(iv) for every $n, m, k \in \mathbb{N}$ (with $k \geq 2$), every $s \in \Sigma^{\circledast}[n, m]$, and every $t_1, \ldots, t_k \in \Sigma^{\circledast}[m, 1]$, let $s; \langle t_1, \ldots, t_k\rangle = \langle s; t_1, \ldots, s; t_k\rangle$.

2.7 Example The concatenation of s and t above is
$s; t = \langle\langle\pi^3_2, \pi^3_2\rangle\sigma, \langle\pi^3_1, \langle\rangle_3\alpha\rangle\sigma\rangle\sigma \in \Sigma^{\circledast}[3, 1]$, which is the forest r shown in Fig. 2.

One can easily show that the forest concatenation is associative and that for every $n, m \in \mathbb{N}$ and $s \in \Sigma^{\circledast}[n, m]$ we have that $\langle\pi^n_1, \ldots, \pi^n_n\rangle; s = s = s; \langle\pi^m_1, \ldots, \pi^m_m\rangle$. Hence, we have a category (that we also denote by $\Sigma^{\circledast}$) whose objects are the non-negative integers, the terminal object is 0, the cartesian product is given by the usual addition, and the π^n_i are the projections. I will write ε_n for $\langle\pi^n_1, \ldots, \pi^n_n\rangle \in \Sigma^{\circledast}[n, n]$ if $n \geq 1$ (resembling the empty word ε in the free monoid) and ε_0 for $\langle\rangle_0 \in \Sigma^{\circledast}[0, 0]$.

3 Recognizable Forest Languages

One can define recognizability either via automata or via the inverse image of a morphism into a finite structure. In our case this would be a cartesian category with countably many objects but with finite hom-sets. Usually there is a straightforward translation between such a morphism and a *deterministic* finite-state automaton. Our case is no different in that respect. But many costructions on automata, for example concatenation, require the non-deterministic model. Thus, we define the recognizable forest languages via *(non-deterministic) finite-state forest automata* (*fsfa*) which accept those languages. Morally, an fsfa is the same as a *finite-state tree automaton* (*fsta*), see e.g. [GS97], but technically, there is a subtle difference: fsfa have initial as well as final states, whereas fsta do only have final states. On the other hand, usual *finite-state automata* (*fsa*) over words do also have initial as well as final states. Thus, forest automata regain a symmetry which was lost for tree automata. In the definitions that follow, we will stay as close as possible to the theory of fsa for word languages [Per90]. The main difference 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.

3.1 Definition Let $n, m \in \mathbb{N}$. 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*. Since Σ is finite, the relation E_k is empty for almost all $k \in \mathbb{N}$.

In order to define the language accepted by an fsfa $\mathfrak{A}$, it is necessary to extend the relations E_k from symbols in Σ to forests in $\Sigma^{\circledast}$. For this let us define for every $k, l \geq 0$ a relation $E^{\mathfrak{A}}_{k,l} \subseteq (\mathfrak{P}(Q))^k \times \Sigma^{\circledast}[k,l] \times Q^l$ (where $\mathfrak{P}(Q)$ denotes the powerset of Q). Informally speaking, we have $(\langle Q_1, \ldots, Q_k\rangle, t, \langle p_1, \ldots, p_l\rangle) \in E^{\mathfrak{A}}_{k,l}$ iff the forest $t \in \Sigma^{\circledast}[k,l]$ can cause a transformation from the states in $\langle Q_1, \ldots, Q_k\rangle$ to the state tuple $\langle p_1, \ldots, p_l\rangle$. The formal definition is given inductively on the structure of the elements of $\Sigma^{\circledast}[k,l]$.

(i) For every $k \geq 0$ and $Q_1, \ldots, Q_k \subseteq Q$, we have $(\langle Q_1, \ldots, Q_k\rangle, \langle\rangle_k, \langle\rangle) \in E^{\mathfrak{A}}_{k,0}$, where $\langle\rangle_k \in \Sigma^{\circledast}[k,0]$ is the unique $(k,0)$-forest (cf. Def. 2.3) and $\langle\rangle \in Q^0$ is the empty tuple of states,

(ii) for all $k \geq 0$ and $Q_1, \ldots, Q_k \subseteq Q$ and $i \in \{1, \ldots, k\}$ and $q \in Q_i$, we have $(\langle Q_1, \ldots, Q_k\rangle, \pi^k_i, q) \in E^{\mathfrak{A}}_{k,1}$,

(iii) for all $k, k' \geq 0$ and $t \in \Sigma^{\circledast}[k,k']$ and $\sigma \in \Sigma_{k'}$ and $Q_1, \ldots, Q_k \subseteq Q$ and $p_1, \ldots, p_{k'}, q \in Q$, if $(\langle Q_1, \ldots, Q_k\rangle, t, \langle p_1, \ldots, p_{k'}\rangle) \in E^{\mathfrak{A}}_{k,k'}$ 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 all $k \geq 0$ and $l \geq 2$ and $Q_1, \ldots, Q_k \subseteq Q$ and $p_1, \ldots, p_l \in Q$ and $t_1, \ldots, t_l \in \Sigma^{\circledast}[k,1]$, if $(\langle Q_1, \ldots, Q_k\rangle, t_i, p_i) \in E^{\mathfrak{A}}_{k,1}$, for every $i \in \{1, \ldots, l\}$, then $(\langle Q_1, \ldots, Q_k\rangle, \langle t_1, \ldots, t_l\rangle, \langle p_1, \ldots, p_l\rangle) \in E^{\mathfrak{A}}_{k,l}$.

3.2 Definition Let $n, m \in \mathbb{N}$ and $\mathfrak{A} = \langle Q, \Sigma, I, F, E\rangle$ be an (n,m)-fsfa. Then $L(\mathfrak{A}) = \{\, t \in \Sigma^{\circledast}[n,m] \mid \exists \langle p_1, \ldots, p_m\rangle \in F \,.\, (\langle I_1, \ldots, I_n\rangle, t, \langle p_1, \ldots, p_m\rangle) \in E^{\mathfrak{A}}_{n,m} \,\}$ is the language *accepted* (or *recognized*) by $\mathfrak{A}$.

3.3 Example Let Σ be as before. Define the $(0,1)$-fsfa $\mathfrak{A} = \langle Q, \Sigma, I, F, E\rangle$ by $Q = \{p, f\}$ and $I = \emptyset$ and $F = \{f\}$ and

$$E_0 = \{(\langle\rangle, \alpha, p), (\langle\rangle, \alpha, f)\}\,, \quad E_1 = \{(\langle p\rangle, \gamma, p)\}\,, \quad E_2 = \{(\langle p, f\rangle, \sigma, f)\}\,.$$

Then $L(\mathfrak{A})$ is the set of all forests with the shape shown on the right of Fig. 1.

3.4 Definition 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 $\mathsf{Rec}(\Sigma^{\circledast})[n,m]$. Further,

$$\mathsf{Rec}(\Sigma^{\circledast}) = \bigcup_{n,m \in \mathbb{N}} \mathsf{Rec}(\Sigma^{\circledast})[n,m]$$

is the set of all recognizable forest languages over Σ.

The concept of fsta is usually introduced by means of term rewriting systems [GS97,Eng75]. It should be obvious that such an fsta is the same as a $(0,1)$-fsfa (Def. 3.1): 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, since there is not much difference between the tuple $(\langle q_1, \ldots, q_k\rangle, \sigma, q) \in E_k$ and the rewrite rule $\sigma(q_1, \ldots, q_k) \to q$ (with $\sigma \in \Sigma_k$). This means that for a given Σ, the set $\mathsf{Rec}(\Sigma^{\circledast})[0,1]$ is isomorphic to the set of recognizable tree languages over Σ. Sometimes the discussion is casted in terms of tree languages over a ranked alphabet Σ and a finite set of variables X. In [GS97], the set of recognizable tree languages over Σ and X is denoted by $\mathsf{Rec}(\Sigma, X)$. If $n = |X|$ is the number of symbols in X, then $\mathsf{Rec}(\Sigma, X)$ is isomorphic to $\mathsf{Rec}(\Sigma^{\circledast})[n,1]$.

3.5 Definition 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 $\mathfrak{A}$, the relations $E_k \subseteq Q^k \times \Sigma_k \times Q$ are graphs of functions $\delta_k : Q^k \times \Sigma_k \to Q$. In this case the definitions can be simplified by the use of functions $\delta^{\mathfrak{A}}_{k,l} : Q^k \times \Sigma^{\circledast}[k,l] \to Q^l$ instead of the relations $E^{\mathfrak{A}}_{k,l} \subseteq (\mathfrak{P}(Q))^k \times \Sigma^{\circledast}[k,l] \times Q^l$. The functions $\delta^{\mathfrak{A}}_{k,l}$ define a functor from the free cartesian category $\Sigma^{\circledast}$ to a cartesian category with finite hom-sets. More explicitely, for a determinisic $(0,1)$-fsfa, the function $\delta^{\mathfrak{A}}_{0,1} : \Sigma^{\circledast}[0,1] \to Q$ is

a Σ-algebra homomorphism whose inverse image of $F \subseteq Q$ is $L(\mathfrak{A}) \subseteq \Sigma^{\circledast}[0,1]$, as it is in the definition of deterministic tree automata [GS97]. We have the following proposition, whose proof uses the well-known power set construction.

3.6 Proposition *Let $n, m \in \mathbb{N}$ and $\mathfrak{A}$ be an (n,m)-fsfa. Then there exists a deterministic and totally defined (n,m)-fsfa $\mathfrak{A}'$ such that $L(\mathfrak{A}') = L(\mathfrak{A})$.*

In the view of this, the definition of $E^{\mathfrak{A}}_{n,m}$ could be replaced by the much simpler construction in the deterministic automaton. However, as in the case of automata over words, the construction in the proof of the Kleene theorem requires non-deterministic automata and a properly defined state transition relation, which is the reason for not using rewriting rules.

4 Closure Properties of Recognizable Forest Languages

As mentioned before, the set $\mathsf{Rec}(\Sigma^{\circledast})[n,1] \subseteq \mathfrak{P}(\Sigma^{\circledast}[n,1])$ of recognizable $(n,1)$-forest languages over Σ is isomorphic to the set $\mathsf{Rec}(\Sigma, X) \subseteq \mathfrak{P}(T_\Sigma(X))$, where $|X| = n$, of recognizable tree languages [GS97]. Hence, $\mathsf{Rec}(\Sigma^{\circledast})[n,1]$ is closed under the boolean operators intersection, union, and complement.

4.1 Proposition *Let Σ be given and $n \in \mathbb{N}$. If $L_1, L_2 \in \mathsf{Rec}(\Sigma^{\circledast})[n,1]$, then $L_1^C = \Sigma^{\circledast}[n,1] \setminus L_1$ as well as $L_1 \cup L_2$ and $L_1 \setminus L_2$ are recognizable.*

4.2 Proposition *Let Σ be a ranked alphabet and $n, m \in \mathbb{N}$. If $L_1, L_2 \in \mathsf{Rec}(\Sigma^{\circledast})[n,m]$, then $L_1 \cap L_2 \in \mathsf{Rec}(\Sigma^{\circledast})[n,m]$.*

It is important to notice, that the set $\mathsf{Rec}(\Sigma^{\circledast})[n,m]$ is in general not closed under union. However, the recognizable languages are closed under cartesian product. With this as a base, we can define another operation $\uplus$ which will take the place of the union.

4.3 Definition If $L_1 \subseteq \Sigma^{\circledast}[n,m_1]$ and $L_2 \subseteq \Sigma^{\circledast}[n,m_2]$, then

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

is the *cartesian product* of L_1 and L_2.

4.4 Proposition *Let Σ be a ranked alphabet and $n, m \in \mathbb{N}$. Further, let $L \subseteq \Sigma^{\circledast}[n,m]$. Then $L \in \mathsf{Rec}(\Sigma^{\circledast})[n,m]$ iff there are $L_1, \ldots, L_m \in \mathsf{Rec}(\Sigma^{\circledast})[n,1]$, such that $L = L_1 \times \ldots \times L_m$.*

Note that this implies that $\mathsf{Rec}(\Sigma^{\circledast})[n,m] = \mathsf{Rec}(\Sigma^{\circledast})[n,1]^m$.

4.5 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.

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, $\nu; L_2$ stands for $\{\nu\}; L_2$.

4.6 Definition Let $L, L' \subseteq \Sigma^{\circledast}[n,m]$. 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'. Similarly, for a family $\{L_i \mid i \in I\}$ of languages with $L_i \subseteq \Sigma^{\circledast}[n,m]$ for all $i \in I$, define

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

4.7 Proposition *Let Σ be a ranked alphabet, and $n, m \in \mathbb{N}$. If $L, L' \in \mathsf{Rec}(\Sigma^{\circledast})[n,m]$, then $L \uplus L' \in \mathsf{Rec}(\Sigma^{\circledast})[n,m]$.*

4.8 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:

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

If $L' \subseteq \Sigma^{\circledast}[m,k]$, then $L \cdot L' = \biguplus_{t \in L'} L \cdot t$ is the *concatenation* of L and L'.

The construction of the forest concatenation might seem unnatural, but the usual tree concatenation [TW68,GS97] is defined in a similar way: Different occurrences of the same 0-ary symbol can be replaced by different trees. Although the recognizable forest languages are not closed under the naive concatenation, they are closed under concatenation. This means that also $\mathsf{Rec}(\Sigma^{\circledast})$ forms a cartesian category (see also [BÉ93]).

4.9 Proposition *Let Σ be a ranked alphabet, and $n, m, k \in \mathbb{N}$.*
If $L_1 \in \mathsf{Rec}(\Sigma^{\circledast})[n,m]$ and $L_2 \in \mathsf{Rec}(\Sigma^{\circledast})[m,k]$, then $L_1 \cdot L_2 \in \mathsf{Rec}(\Sigma^{\circledast})[n,k]$.

For the proof we need the concept of *normalized* fsfa, where there is only one input state tuple and only one output state tuple and there are no transitions from a final state or into an initial state. Then, the two normalized fsfa for L_1 and L_2 are connected.

Now we can define a generalization of Kleene's star operation as a closure operation for the concatenation. For the concatenation operation of a free monoid this does not bring any problems. Note that this makes sense only for languages $L \subseteq \Sigma^{\circledast}[n,n]$, and we do not take the union of all L^z for $0 \leq z < \omega$ (as it is done for word languages) but use the operation $\biguplus$ instead.

4.10 Definition For $L \subseteq \Sigma^{\circledast}[n,n]$ define

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

This star operation is a generalization of both, Kleene's star operation for word languages and the star operations for trees defined in [TW68].

4.11 Proposition *If* $L \in \mathsf{Rec}(\Sigma^{\circledast})[n,n]$*, then* $L^\star \in \mathsf{Rec}(\Sigma^{\circledast})[n,n]$.

The basic idea of the proof is to construct the normalized fsfa $\mathfrak{A}'$ for L and identify initial and final states. However, note that the situation is not entirely trivial because in general $L(\mathfrak{A}') \neq L$. The reason for this is that the forests containing π_i^n are removed in $L(\mathfrak{A}')$ because otherwise $\mathfrak{A}'$ would not be normal. We have to construct another automaton $\mathfrak{A}''$, which reintroduces these forests.

5 Rational Forest Languages

The set of rational forest languages is the smallest set that contains the finite languages and that is closed under componentwise union, cartesian product, concatenation, and the star operation.

5.1 Definition Let Σ be a ranked alphabet. For every $n, m \in \mathbb{N}$, the set $\mathsf{Rat}(\Sigma^{\circledast})[n,m]$ is the smallest set such that

- $\emptyset \in \mathsf{Rat}(\Sigma^{\circledast})[n,m]$, and if $t \in \Sigma^{\circledast}[n,m]$, then $\{t\} \in \mathsf{Rat}(\Sigma^{\circledast})[n,m]$,
- if $L_1, L_2 \in \mathsf{Rat}(\Sigma^{\circledast})[n,m]$, then $L_1 \uplus L_2 \in \mathsf{Rat}(\Sigma^{\circledast})[n,m]$,
- if $L_1 \in \mathsf{Rat}(\Sigma^{\circledast})[n,m_1]$ and $L_2 \in \mathsf{Rat}(\Sigma^{\circledast})[n,m_2]$, then $L_1 \times L_2 \in \mathsf{Rat}(\Sigma^{\circledast})[n,m_1+m_2]$,
- if $L_1 \in \mathsf{Rat}(\Sigma^{\circledast})[n,k]$ and $L_2 \in \mathsf{Rat}(\Sigma^{\circledast})[k,m]$, then $L_1 \cdot L_2 \in \mathsf{Rat}(\Sigma^{\circledast})[n,m]$,
- if $L \in \mathsf{Rat}(\Sigma^{\circledast})[n,n]$, then $L^\star \in \mathsf{Rat}(\Sigma^{\circledast})[n,n]$.

The set of all rational forest languages over Σ is defined as

$$\mathsf{Rat}(\Sigma^{\circledast}) = \bigcup_{n,m \in \mathbb{N}} \mathsf{Rat}(\Sigma^{\circledast})[n,m] \quad .$$

From the propositions in the previous section it follows that for every ranked alphabet Σ and $n, m \in \mathbb{N}$ we have that $\mathsf{Rec}(\Sigma^{\circledast})[n,m] \subseteq \mathsf{Rat}(\Sigma^{\circledast})[n,m]$. The following theorem says that the converse is also true.

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

In the proof of this theorem, the rational expression is constructed inductively on the size of the fsfa. The construction is inspired by Kleene's original construction [Kle56], but it is quite different from the one used by Thatcher and Wright [TW68]. In particular, note that *a priori* there is no relation between $\mathsf{Rat}(\Sigma^{\circledast})[n,1]$ and the rational languages of [TW68]. Hence, Theorem 5.2 is not a consequence of the result in [TW68]. But from Theorem 5.2 and [TW68] it follows that $\mathsf{Rat}(\Sigma^{\circledast})[n,1]$ coincides with the rational languages of [TW68].

5.3 Example For the forest language recognized by the automaton in Example 3.3 we can obtain the rational expression depicted in Fig. 3:

$$L = \{\langle\rangle_0\alpha\} \uplus \big(\{\langle\rangle_0\alpha\} \cdot \{\pi_1^1\gamma\}^\star \times \{\langle\rangle_0\alpha\}\big) \cdot \{\langle\pi_1^2, \langle\pi_1^2, \pi_2^2\rangle\sigma\rangle\}^\star \cdot \{\langle\pi_1^2, \pi_2^2\rangle\sigma\} \quad .$$

By using Notation 2.5, we can write this more concisely as

$$L = \{\alpha\} \uplus (\{\alpha\} \cdot \{\gamma\}^\star \times \{\alpha\}) \cdot \{\langle\pi_1^2, \sigma\rangle\}^\star \cdot \{\sigma\} \quad .$$

Now we can immediately derive the main theorem of this paper.

5.4 Theorem *Let* Σ *be a ranked alphabet. Then* $\mathsf{Rat}(\Sigma^{\circledast}) = \mathsf{Rec}(\Sigma^{\circledast})$.

6 Conclusions and Future Work

The algebraic concepts used in this paper have been studied for a long time (e.g., [Law63,Ésik81,AD82,BÉ93]), but the construction of rational expression for the Kleene theorem (Theorem 5.4) in this paper has (up to my knowledge) not yet appeared in the literature. Apart from the independent interest of the result, the work is motivated by the following issues of future research.

- The original motivation for proposing this new approach comes from the desire to give algebraic characterizations for certain classes of tree transducers (e.g., [Eng75,FV98]). Their counterparts for word languages, e.g., general seqential machines [GR66], have been algebraicly characterized by using Kleene's original theorem (see e.g. [Ber79]). For tree transductions it is more difficult to find such characterizations because of the insufficient behaviour of the tree concatenation operation (see also [AD82]). The here proposed notion of rational expression allows to give an algebraic characterization of the transductions realized by sequential forest transducers [Str00], which are a generalization of a certain class of bottom-up tree transducers. In [AD82], Arnold and Dauchet encountered the same problem when studying bimorphisms for trees. Their solution uses *magmoides*. The structures $\Sigma^{\circledast}$ and $\mathsf{Rec}(\Sigma^{\circledast})$ discussed in this paper are magmoides (see also [ÉW05]).
- In [DPV05], the result of [TW68] is generalized to trees over arbitrary semirings (as done in [Sch61] for word automata). This raises the question whether also Theorem 5.4 can be proved for forests over arbitrary semirings.
- Another important problem is whether the result can be extended to unranked trees and hedges [Cou89,CDG$^+$07] which are important for practical applications [Nev02,MN07]. The notion of cartesian category already provides the right algebraic structure. We only have to add additional objects (infinite ordinals) to the category of forests.
- More generally, it is now possible to generalize the notion of recognizable and rational languages from the free cartesian category to any cartesian category, in the same sense as it had been generalized from the free monoid to any monoid [Eil76,Ber79]. The question is in which cases Theorem 5.4 holds.
- The same question can be raised for the definability in monadic second order logic [TW68,Tho97,Nev02,CDG$^+$07].

References

[AD82] A. Arnold and M. Dauchet. Morphisms et bimorphisms d'arbres. *Theoretical Computer Science*, 20:33–93, 1982.

[BÉ93] S. L. Bloom and Z. Ésik. *Iteration Theories*. Springer-Verlag, 1993.

[Ber79] Jean Berstel. *Transductions and Context-Free Languages*. Number 38 in Leitfäden der angewandten Mathematik und Mechanik LAMM. B.G. Teubner Stuttgart, 1979.

[Boj07] Mikolaj Bojańczyk. Forest expressions. In *CSL'07*, volume 4646 of *Lecture Notes in Computer Science*, pages 146–160. Springer-Verlag, 2007.

[CDG$^+$07] H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications, 2007. release October, 12th 2007.

[Cou89] Bruno Courcelle. On recognizable sets and tree automata. In M. Nivat and H. Ait-Kaci, editors, *Resolution of equations in algebraic structures*, pages 93–126. Academic Press, 1989.

[DG99] Manfred Droste and Paul Gastin. The Kleene–Schützenberger theorem for formal power series in partially commuting variables. *Information and Computation*, 153:47–80, 1999.

[DPV05] Manfred Droste, Christian Pech, and Heiko Vogler. A Kleene theorem for weighted tree automata. *Theory Comput. Syst.*, 38(1):1–38, 2005.

[Ede85] Elmar Eder. Properties of substitutions and unifications. *Journal of Symbolic Computation*, 1(1):31–46, 1985.

[Eil76] Samuel 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.

[Ési81] Zoltán Ésik. An axiomatization of regular forests in the language of algebraic theories with iteration. In *FCT*, volume 117 of *Lecture Notes in Computer Science*, pages 130–136. Springer-Verlag, 1981.

[ÉW05] Zoltán Ésik and Pascal Weil. Algebraic recognizability of regular tree languages. *Theoretical Computer Science*, 340:291–321, 2005.

[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] Stephen Cole 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.

[Law63] F. W. Lawvere. Functorial semantics of algebraic theories. In *Proceedings of the National Academy of Sciences, USA*, volume 50, pages 869–872. National Academy of Sciences, 1963. Summary of PhD thesis, Columbia University.

[MN07] W. Martens and J. Niehren. On the minimization of XML schemas and tree automata for unranked trees. *J. Comput. Syst. Sci.*, 73(4):550–583, 2007.

[Nev02] Frank Neven. Automata, logic, and XML. In *CSL'02*, volume 2471 of *LNCS*, pages 2–26. Springer-Verlag, 2002.

[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.

[Sch61] M. P. Schützenberger. On the definition of a family of automata. *Inform. And Control*, 4:245–270, 1961.

[Str00] Lutz Straßburger. Rational forest languages and sequential forest transducers. Master's thesis, Technische Universität Dresden, 2000.

[Tho97] Wolfgang Thomas. Languages, automata, and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Language Theory*, volume 3, pages 389–456. Springer-Verlag, 1997.

[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.