Text Languages in an Algebraic Framework*

Hendrik Jan HOOGEBOOM
Paulien ten PAS

Department of Computer Science, Leiden University,
P.O.Box 9512, 2300 RA Leiden, The Netherlands
hjh@rulwinw.leidenuniv.nl, pas@rulwinw.leidenuniv.nl

Abstract. A text can be defined as a word w together with a (second) linear order on its domain $\{1, ..., |w|\}$. This second order may be used to define a hierarchical, tree-like, structure representing the text. The family of *context-free* sets of texts is investigated, i.e., sets of texts defined by context-free text grammars. In particular, those sets of texts are studied in the framework of *universal algebra*. This allows to compare the classical notions of *equational* and *recognizable* families in an algebra with context-free sets in the "algebra of texts". Within this algebra the notion of equational sets coincides with the context-free sets. A grammatical characterization of the family of recognizable sets is given as a subfamily of the context-free sets of texts.

1 Introduction

This paper further investigates the class of context-free texts, that was introduced in [9] generalizing context-free word grammars to the setting of texts.

The notion of a text itself generalizes the notion of a word. A *text* is a triple $\tau = (\lambda, \rho_1, \rho_2)$ such that λ is a labeling function from a domain D to some alphabet, and ρ_1 and ρ_2 are linear orders on the domain D of λ. Usually, $D = \{1, 2, \ldots, n\}$ and ρ_1 is of the standard form $(1, 2, \ldots, n)$. Hence τ may be seen as a word $\lambda(1)\lambda(2)\cdots\lambda(n)$ (referred to as the *word of* τ) together with an additional linear order ρ_2 on the domain $\{1, \ldots, n\}$ of λ.

The traditional role of a context-free word grammar is to define a language as a set of words (generated by the grammar) and to provide each word in the language with its syntactic structure (given by a derivation tree of the grammar). In the case of texts, each text already has an *intrinsic* tree-like structure, called its *shape*, a notion which originates from the decomposition theory of 2-structures (see [5, 6]). Hence, rather than providing each text with a syntactic structure, the role of a context-free text grammar is limited to that of defining a set of texts.

The tree-like structure used to represent a text hierarchically is a so-called leaf-labeled *bi-ordered* tree, which generalizes the concept of a leaf-labeled ordered tree

*Research supported by the EBRA Working Group ASMICS 2.

giving a structure to a word. A tree is bi-ordered if with each inner node *two* linear orderings of its children are associated. These local orderings then determine two orderings on the leaves of the tree.

If the second order of a text equals the first order or the reverse of the first order, then such a *forward* or *backward sequential* text is very much like a word: the text does not impose any structure on the bi-ordered tree representation. On the other hand, very unlike the case of words, there is an important class of texts that have only one (rather trivial) representation, where the leaves of the tree are children of the root, and the associated orderings of the root are the orderings of the text itself. These text are called *primitive*.

A bi-ordered tree representation for a text describes a modular decomposition of the text. This way of decomposing texts is in complete accordance with the decomposition theory for 2-structures (see [7]), due to a close correspondence between texts and a certain subclass of 2-structures. Primitive texts, being undecomposable, play the role of primes in this theory.

It is natural to look for a "maximal" decomposition of a text, i.e., a tree representation where all nodes have a primitive structure. Such a maximal decomposition may not be unique, as in the case of sequential texts where many binary bi-ordered tree representations for each text exist.

However, it turns out that maximal decompositions of a text differ only in their binary subtrees representing sequential parts of the text. Now the *shape* of a text is the unique bi-ordered tree representing the text such that each inner node is either primitive or sequential, and for each forward (backward) sequential node, none of its children is forward (backward, respectively) sequential. Thus, the shape indicates how the text is built up with words (the sequential nodes) and primitive building blocks. Every maximal decomposition of a text can be obtained from its shape by decomposing each sequential node into a binary (bi-ordered) subtree.

Context-free text grammars are direct generalizations of context-free word grammars. In [9] this class of grammars and their text languages were studied.

A text on the one hand can be seen as a *word* with an additional ordering, on the other hand it has an intrinsic *tree*-like structure (as, e.g., given by its shape). The main motivation of this paper is to explore this apparent duality. In particular we want to compare and relate the families of context-free word languages, context-free text languages, and tree languages generated by "regular" tree grammars.

The natural framework to relate these three different structures is that of universal algebra. An algebra is a set A together with a collection Σ of operations on A. If A is the set of words over a given alphabet, then Σ contains only concatenation. Choosing A to be a family of (ranked, ordered) trees, then an operation of rank n in Σ builds a tree out of n given subtrees. For texts one may take operations corresponding to the primitive (de-)compositions of texts.

In this algebra of texts we study the well-established notion of *equational* languages, that formalizes (in an algebraic setting) the notion of a language specified by a set of recursive equations (a context-free grammar can be interpreted as such). Using elementary techniques we show that the equational text languages coincide with the context-free text languages. Additionally we consider the *recognizable* sets, which extends the idea of languages accepted by a finite state device.

For text languages, as for word languages, the recognizable sets are strictly included in the equational sets. For tree languages the two notions coincide.

We isolate a class of context-free grammars that precisely generate the recognizable text languages. We call a context-free text grammar *right-linear* if its (bi-ordered) derivation trees are "right-most" maximal decompositions of the derived texts. (Such a right-most maximal decomposition is obtained from the shape by decomposing each sequential node into a "right-most" binary subtree; thus in this way each text has a unique right-most maximal decomposition.) Hence, derivation trees are allowed to be "regular tree-like" where the shape has primitive nodes, but they are restricted to "right-linear" sequential parts. This class of right-linear grammars is more powerful than the *shapely* grammars from [9], which allow only shapes as derivation trees.

The paper is organized as follows. We start by giving some preliminaries. In Section 3 we present the framework of universal algebra with the classical definitions of recognizable and equational subsets of an algebra. Additionally, in Section 4 we recall the basic notions and results on texts and their hierarchical representations.

In Section 5 we provide an algebraic framework for texts, and start considering the recognizable and equational text languages. The equational text languages are then the context-free text languages from [9].

In Section 6, we give characterizations of recognizable text languages. In particular we prove that the recognizable text languages are precisely those text languages generated by the so-called right-linear text grammars (Theorem 6.9). In Section 7 we consider how the notions of recognizability and equationality for text languages are related to those for word languages and tree languages. Finally, in Section 8 we consider some closure properties of families of text languages.

2 Preliminaries

For a (finite) sequence $s = (x_1, \ldots, x_n)$, $|s|$ denotes its length n, and for $1 \leq i \leq |s|$, $s(i)$ denotes the i'th element x_i of s. In particular, we view a word w over an alphabet Δ as a sequence of letters of Δ, but as usual we write $w = a_1 \cdots a_n$ if $w(i) = a_i \in \Delta$ for $i = 1, \ldots, n$.

For a non-empty finite set D, a *linear order (on D)* is a relation ρ on D such that ρ is antireflexive, transitive, and total, i.e., for all $x, y \in D$ with $x \neq y$, either $(x, y) \in \rho$ or $(y, x) \in \rho$. For each linear order ρ on D there is a unique ordering $x_1, \ldots, x_n$ of the elements in D such that $(x_i, x_j) \in \rho$ iff $i < j$. Hence a linear order ρ on D can be specified as a sequence of the elements $(x_1, \ldots, x_n)$ of D. The terminology and notations concerning sequences carry over to linear orders.

For a linear order $\rho = (x_1, \ldots, x_n)$, we use $dom(\rho)$ to denote the set $\{x_1, \ldots, x_n\}$. A subset $X \subseteq dom(\rho)$ is a *segment of* ρ if there exist $i, j \in \{1, \ldots, n\}$ such that $X = \{x_\ell \mid i \leq \ell \leq j\}$ (this includes $X = \emptyset$).

For linear orders $\rho_1 = (x_1, \ldots, x_n)$ and $\rho_2 = (y_1, \ldots, y_m)$ with disjoint domains, the *sum of* ρ_1 and ρ_2, denoted $\rho_1 + \rho_2$, is the linear order specified by the sequence $(x_1, \ldots, x_n, y_1, \ldots, y_m)$. Note that this sum operation is not commutative.

If a function f on a set D is given, then we shall extend f in the usual way to a subset X of D, yielding the set $f(X)$, or to a sequence ρ on D, yielding the sequence $f(\rho)$.

By a *tree* t we mean a directed graph with one designated node, the *root of* t, such that each node is connected with the root by a unique directed path from the root. The nodes without outgoing edges are the *leaves* of t, the other nodes are the *inner* nodes of t. A tree is *chain-free* if it has no nodes with precisely one outgoing edge. The *out-degree* of t is the maximum number of outgoing edges per node. A *node-labeled* (or *inner-*, *leaf-labeled*) tree is a tree where in addition each node (or each inner node or each leaf) has a label.

An *ordered tree* is a tree together with a function *ord* that associates to each inner node v a linear order $ord(v)$ on the children of v. These local linear orders induce a linear order ρ on the leaves of the tree. The *yield* of a leaf-labeled ordered tree is the word formed by the labels of the leaves according to the induced ordering of the leaves.

For our purposes, the identity of the nodes of trees and ordered trees is not important. Hence we will consider (ordered) trees modulo the identity of their nodes.

A context-free grammar G is denoted by a 4-tuple (N, Δ, P, S), where N is the alphabet of nonterminals, Δ is the alphabet of terminals, P is the set of productions of the form $A \to w$ with $A \in N$ and $w \in (N \cup \Delta)^*$, and $S \in N$ is the axiom. To emphasize the fact that G is used to generate words we call it a context-free *word* grammar.

3 Sigma-algebras

We recall here some notions concerning universal algebra — see, e.g., [1]. A *ranked alphabet* Σ is a finite alphabet of operator symbols, where each operator symbol $\sigma \in \Sigma$ has a rank $r(\sigma) \in \mathbf{N}$; for $m \in \mathbf{N}$, Σ_m denotes $\{\sigma \in \Sigma \mid r(\sigma) = m\}$. A Σ-*algebra* $\mathcal{A}$ is a pair (A, Σ), where A is a set and Σ a ranked alphabet, and each operator $\sigma \in \Sigma_m$, $m \geq 0$, defines a mapping $\sigma^{\mathcal{A}} : A^m \to A$.

Let Σ be a fixed ranked alphabet.

Let $\mathcal{A} = (A, \Sigma)$ and $\mathcal{B} = (B, \Sigma)$ be Σ-algebras. A *homomorphism* h from $\mathcal{A}$ to $\mathcal{B}$ is a mapping $h : A \to B$ such that for each $\sigma \in \Sigma_m$, $h(\sigma^{\mathcal{A}}(a_1, \ldots, a_m)) = \sigma^{\mathcal{B}}(h(a_1), \ldots, h(a_m))$ for all $a_1, \ldots, a_m \in A$. A *congruence of* $\mathcal{A}$ is a relation on A that is invariant under every operator, i.e., $\cong$ is a congruence if, for all $\sigma \in \Sigma_m$, and all $a_1, \ldots, a_m, a'_1, \ldots, a'_m \in A$, $a_i \cong a'_i$ for $1 \leq i \leq m$ implies $\sigma^{\mathcal{A}}(a_1, \ldots, a_m) \cong \sigma^{\mathcal{A}}(a'_1, \ldots, a'_m)$. An *elementary translation of* $\mathcal{A}$ is a mapping $\varphi : A \to A$ defined by $\varphi(v) = \sigma^{\mathcal{A}}(a_1, \ldots, a_{j-1}, v, a_{j+1}, \ldots, a_m)$, where $\sigma \in \Sigma_m$, $1 \leq j \leq m$, and $a_1, \ldots, a_{j-1}$, $a_{j+1}, \ldots, a_m \in A$. A *translation of* $\mathcal{A}$ is the composition of elementary translations of $\mathcal{A}$. A relation on A is a congruence of $\mathcal{A}$ iff it is invariant under the (elementary) translations of $\mathcal{A}$.

Given a congruence $\cong$ of $\mathcal{A}$, the corresponding *quotient algebra*, denoted by $\mathcal{A}/\cong$, is the Σ-algebra (C, Σ), where C consists of the congruence classes of $\cong$, and for $\sigma \in \Sigma_m$, $\sigma^{\mathcal{A}/\cong}(c_1, ..., c_m)$ is the class of $\sigma^{\mathcal{A}}(a_1, ..., a_m)$, where a_i is a representative of c_i for $i = 1, \ldots, m$.

Homomorphisms and congruences are related as follows : the kernel of a homomorphism h from $\mathcal{A}$ to $\mathcal{B}$, denoted by $\ker(h)$, is the congruence such that $a, a' \in A$ are congruent iff $h(a) = h(a') \in B$, and each congruence $\cong$ of $\mathcal{A}$ is the kernel of the homomorphism from $\mathcal{A}$ to $\mathcal{A}/\cong$ which assigns to each element of A its congruence class.

Let V be a set of variables. The set of ΣV-*terms*, denoted by $F_\Sigma(V)$, is the smallest set of words over $\Sigma \cup V \cup \{\langle, \rangle\}$ that contains $V \cup \Sigma_0$, and such that if $m \geq 1$, $t_1, \ldots, t_m \in F_\Sigma(V)$, $\sigma \in \Sigma_m$, then $\sigma\langle t_1 \cdots t_m \rangle \in F_\Sigma(V)$. For $V = \emptyset$, we denote $F_\Sigma(\emptyset)$ by F_Σ. Note that $F_\Sigma = \emptyset$ iff $\Sigma_0 = \emptyset$. By considering the variables in V as nullary symbols in Σ we identify $F_\Sigma(V)$ with $F_{\Sigma \cup V}$.

The Σ-algebra of (ground) *terms* $\mathcal{F}_\Sigma = (F_\Sigma, \Sigma)$ is the Σ-algebra such that $\sigma^{\mathcal{F}_\Sigma}(t_1, \ldots, t_m) = \sigma\langle t_1 \cdots t_m \rangle$, for $\sigma \in \Sigma_m$ with $m \geq 1$, $t_1, \ldots, t_m \in F_\Sigma$, and $\sigma^{\mathcal{F}_\Sigma} = \sigma$ for $\sigma \in \Sigma_0$. $\mathcal{F}_\Sigma$ is initial in the class of all Σ-algebras, i.e., for each Σ-algebra $\mathcal{A}$ there is a unique homomorphism from $\mathcal{F}_\Sigma$ to $\mathcal{A}$; if this homomorphism is surjective we say that $\mathcal{A}$ is *generated by* Σ.

Let $\mathcal{A} = (A, \Sigma)$ be a Σ-algebra, and let $\{v_1, \ldots, v_n\}$ be an (ordered) set of variables. With each term $t \in F_\Sigma(\{v_1, \ldots, v_n\})$ we associate a mapping $t^{\mathcal{A}} : A^n \to A$, which is defined by $v_i^{\mathcal{A}}(a_1, \ldots, a_n) = a_i$, and $\sigma\langle t_1 \cdots t_m \rangle^{\mathcal{A}}(a_1, \ldots, a_n) = \sigma^{\mathcal{A}}(t_1^{\mathcal{A}}(a_1, \ldots, a_n)), \ldots$ $\ldots, t_m^{\mathcal{A}}(a_1, \ldots, a_n))$; $t^{\mathcal{A}}$ is a so-called *derived operator*.

A *theory* T is a pair (Σ, E) where Σ is a ranked alphabet, and E is a set of equations of the form $t = t'$ where $t, t' \in F_\Sigma(V)$ for a set of variables V. The Σ-algebra $\mathcal{A} = (A, \Sigma)$ is called a T-*algebra* if it satisfies the equations of T: $t^{\mathcal{A}} = t'^{\mathcal{A}}$ for each $(t = t') \in E$. In particular, the *quotient term algebra* $\mathcal{F}_\Sigma / \cong$, where $\cong$ is the congruence on $\mathcal{A}$ generated by the equations in E, is a T-algebra; moreover it is initial in the class of T-algebras.

3.1 Recognizable and equational sets

We recall the basic notions of recognizable and equational sets, in the setting of Σ-algebras (as introduced in [12], see also, e.g., [2], [4], [15]). Let Σ be an arbitrary, but fixed ranked alphabet, and let $\mathcal{A} = (A, \Sigma)$ and $\mathcal{B} = (B, \Sigma)$ be Σ-algebras.

The notion of word languages recognizable by finite state automata can be generalized to subsets of an algebra. Finite algebras take the role of (deterministic) finite state acceptors, and the mapping that assigns to each word the state reached is in this setting a homomorphism of algebras.

Definition 3.1 A subset $K \subseteq A$ is *recognizable* if there is a finite Σ-algebra $\mathcal{Q} = (Q, \Sigma)$, a homomorphism $h : \mathcal{A} \to \mathcal{Q}$, and a subset $F \subseteq Q$ such that $h^{-1}(F) = K$.

In view of the natural correspondence of homomorphisms and congruences, one might alternatively define that $K \subseteq A$ is recognizable if there is a finite congruence of $\mathcal{A}$ that saturates K (i.e., there are finitely many congruence classes and K is the union of some of them). It is well known that the greatest congruence saturating a set K, called the Nérode congruence of K and denoted by $\cong_K$, can be characterized as follows: $a \cong_K a'$ iff for every translation φ of $\mathcal{A}$, $\varphi(a) \in K$ iff $\varphi(a') \in K$.

Proposition 3.2 *$K \subseteq A$ is recognizable iff $\cong_K$ is finite.* □

Context-free word grammars may be seen as a recursive mechanism for specifying languages. In the framework of universal algebra this generalizes to systems of equations and equational sets.

A *polynomial system* S is a set of equations $\{v_i = t_{i1} + \cdots t_{ik_i} \mid i = 1, \ldots, n\}$, where $\{v_1, \ldots, v_n\}$ is a fixed set of variables, and each t_{ij} is a term in $F_\Sigma(\{v_1, \ldots, v_n\})$. With such a polynomial system S one associates a system function $S^{\mathcal{A}} : (2^A)^n \to$

$(2^A)^n$ satisfying $S^{\mathcal{A}}(W_1,\ldots,W_n) = (W_1',\ldots,W_n')$, where $W_i' = \bigcup_{j=1}^{k_i} t_{ij}^{\mathcal{A}}(W_1,\ldots,W_n)$, for $W_1,\ldots,W_n \subseteq A$. Being a continuous mapping, the system function S has a least fixed point, denoted by $[S^{\mathcal{A}}]$.

Definition 3.3 A subset $K \subseteq A$ is *equational* if there is a polynomial system S such that K is a component of the least solution $[S^{\mathcal{A}}]$.

The next theorem collects some facts concerning the behaviour of homomorphisms with respect to recognizability and equationality.

Theorem 3.4 *Let $h : \mathcal{A} \to \mathcal{B}$ be a homomorphism.*
(1) *If $L \subseteq A$ is equational, then $h(L) \subseteq B$ is equational.*
(2) *If $K \subseteq B$ is recognizable, then $h^{-1}(K) \subseteq A$ is recognizable.*
(3) *If h is surjective, and if $h^{-1}(K) \subseteq A$ is recognizable (equational), then $K \subseteq B$ is recognizable (equational).*
(4) *If h is injective, and if $h(L) \subseteq B$ is recognizable (equational), then $L \subseteq A$ is recognizable (equational).*
(5) *If $K \subseteq B$ is equational, then there exists an equational set $L \subseteq A$ such that $K = h(L)$.*

Proof. The results for equational sets rely on the general fact (see [12]) that each homomorphism preserves the least fixed point of a polynomial system S, i.e., $h[S^{\mathcal{A}}] = [S^{\mathcal{B}}]$, where h is extended to sequences of subsets of A. E.g., claim (5) follows from the fact that if K is the i'th component of $[S^{\mathcal{B}}]$ for some i, S being a polynomial system, then the i'th component of $[S^{\mathcal{A}}]$ is an equational subset of A and its image under h is K.

For recognizable sets, (2) and (4) follow immediately from the fact that if $K = j^{-1}(F)$ for some homomorphism j, then $h^{-1}(K) = (j \circ h)^{-1}(F)$.

The proof of (3) is as follows. Due to the surjectivity of h, for each (elementary) translation ψ of $\mathcal{B}$ there exists an (elementary) translation φ of $\mathcal{A}$ such that $h(\varphi(a)) = \psi(h(a))$ for each $a \in A$. Assuming that $a \cong_{h^{-1}(K)} a'$ for some $a, a' \in A$, then also $h(a) \cong_K h(a')$ by the characterization of the Nérode congruence in terms of translations. Using again the surjectivity of h we may infer that the index of $\cong_K$ is not larger than the index of $\cong_{h^{-1}(K)}$. Consequently, K is recognizable whenever $h^{-1}(K)$ is. □

Note that in particular it follows from this theorem that isomorphisms preserve recognizability and equationality.

In the case of the term Σ-algebra F_Σ, we have the following result from [12].

Proposition 3.5 *For each $T \subseteq F_\Sigma$, T is equational iff T is recognizable.* □

For arbitrary Σ-algebras that are generated by Σ this result holds in only one direction.

Corollary 3.6 *Let $\mathcal{A} = (A, \Sigma)$ be a Σ-algebra generated by Σ. If $K \subseteq A$ is recognizable, then K is equational.*

Proof. Let h be the unique homomorphism from $\mathcal{F}_\Sigma$ to $\mathcal{A}$. Since $\mathcal{A}$ is generated by Σ, h is surjective. By Theorem 3.4(2), $h^{-1}(K) \subseteq F_\Sigma$ is recognizable. By Proposition 3.5, $h^{-1}(K)$ is equational. By Theorem 3.4(1), $h(h^{-1}(K))$ is equational. Since h is surjective, $K = h(h^{-1}(K))$. □

The general concepts given above can again be specialized to word languages and tree languages.

The word languages in this paper are all ε-free, and hence we work in the semi-group Δ^+ rather than in the monoid Δ^*. Any semi-group can be viewed as a Σ-algebra, where Σ consists of one operation of rank 2. In the case of the free semi-group Δ^+, we may extend Σ by adding the elements of Δ as nullary operators, which makes Δ^+ an algebra generated by its ranked alphabet. More precisely, Δ^+ is a Σ-algebra, with $\Sigma = \Delta \cup \{\sigma\}$ such that σ is an operator of rank 2 that is interpreted as word concatenation, and every nullary operator $a \in \Delta$ is interpreted as the word a of length 1. Moreover, the associativity of concatenation can be expressed by stating that Δ^+ satisfies the equation $e : \sigma\langle u\sigma\langle vw\rangle\rangle = \sigma\langle\sigma\langle uv\rangle w\rangle$, i.e., Δ^+ is a T-algebra for the theory $T = (\Sigma, \{e\})$; the freeness of Δ^+ is expressed by stating that it is an *initial* T-algebra. As is well-known (see,e.g., [12]) the recognizable subsets of Δ^+ are then precisely the (ε-free) word languages recognized by finite state automata, and the equational subsets of Δ^+ are precisely the (ε-free) context-free word languages.

For any ranked alphabet Σ, the terms in F_Σ describe node-labeled ordered trees (modulo the identity of their nodes). Accordingly, the subsets of F_Σ are known as *tree languages* (for an overview on tree languages, see, e.g., the book [11]). Recognizable tree languages are usually defined as tree languages accepted by so-called deterministic bottom-up tree recognizers. This definition is equivalent with Definition 3.1.

By Proposition 3.5 a tree language is recognizable iff it is equational. A polynomial system for an equational tree language corresponds closely with the notion of *regular tree grammar*, which is a 4-tuple (N, Δ, P, S), where $\Delta = \Sigma_0$, N is the set of nonterminals disjoint from Δ, $S \in N$, and P consists of productions of the form $A \to t$, where $A \in N$ and $t \in F_\Sigma(N)$. A tree $t' \in F_\Sigma(N)$ is derived from a tree $t \in F_\Sigma(N)$, denoted $t \Rightarrow_G t'$, if there is a production $A \to u$ in P such that t' is obtained from t by substituting the tree u for an occurrence of A. As usual $\Rightarrow_G^*$ denotes the transitive and reflexive closure of $\Rightarrow_G$. The *tree language generated by* the regular tree grammar $G = (N, \Delta, P, S)$, denoted by $\mathrm{TrL}(G)$, is the set of trees $\{t \in F_\Sigma \mid S \Rightarrow_G^* t\}$. A regular tree grammar is *in normal form* if each production is of the form $A \to \sigma\langle A_1 \cdots A_m\rangle$ with $\sigma \in \Sigma_m$, $m \geq 0$, and $A_1, \ldots, A_m \in N$.

A regular tree grammar $H = (N, \Delta, P, A_k)$, where $N = \{A_1, \ldots, A_n\}$, corresponds with a polynomial system S with equations $A_i = t_{i1} + \cdots + t_{ik_i}$, for $i = 1, \ldots, n$, where $A_i \to t_{i1}, \ldots, A_i \to t_{ik_i}$ are the productions in P with A_i as left-hand side. The k'th component of $[S^{\mathcal{F}}]$ equals $\mathrm{TrL}(H)$.

Let us note here that also the notion of "context-free tree grammar" exists (see [13]). In such a grammar non-terminals may have nonzero rank. The class of tree languages generated by these context-free tree grammars strictly contains the class of equational tree languages occurring in this paper.

4 Texts and text languages

All notions and results given in this section are from [5, 6, 7, 9]. We have tried to keep this overview as brief as possible. For additional technical details we refer to the above-mentioned papers.

4.1 Texts and bi-orders

Let Δ be an alphabet.

Definition 4.1 A *text* τ *(over* Δ*)* is a triple $(\lambda, \rho_1, \rho_2)$, where ρ_1 and ρ_2 are linear orders such that $dom(\rho_1) = dom(\rho_2)$, and λ is a function from $dom(\rho_1)$ to Δ.

For a text $\tau = (\lambda, \rho_1, \rho_2)$, the *domain of* τ, denoted by $dom(\tau)$, is $dom(\rho_1)$; the *word of* τ, denoted by $word(\tau)$, is the word $\lambda(\rho_1) \in \Delta^+$.

The pair (ρ_1, ρ_2) determines the structural properties of the text τ. A pair of linear orders $\sigma = (\rho_1, \rho_2)$ such that $dom(\rho_1) = dom(\rho_2)$ is called a *bi-order*; the common domain of ρ_1 and ρ_2 is denoted by $dom(\sigma)$.

Bi-orders (and hence texts) correspond with a specific kind of labeled 2-structures ([7, 9]). As a consequence the decomposition theory of 2-structures has a translation to bi-orders. We give here only the result of this translation, and not the details concerning 2-structures.

For a bi-order $\sigma = (\rho_1, \rho_2)$, a subset $X \subseteq dom(\sigma)$ is a *clan of* σ if X is a segment of both ρ_1 and ρ_2. Note that for each bi-order σ, the subsets $\emptyset$, $dom(\sigma)$, and the singletons in $dom(\sigma)$ are all clans of σ, the so-called *trivial* clans. A bi-order is *primitive* if it has no non-trivial clans; it is *sequential* if all segments of both of its linear orders are clans. There are two possible forms for a sequential bi-order $\sigma = (\rho_1, \rho_2)$: either ρ_1 equals ρ_2 (then σ is called *forward sequential*) or ρ_1 and ρ_2 are reverses of each other (then σ is called *backward sequential*). These notions carry over to texts.

In this paper, we will work with abstract bi-orders and texts, i.e., isomorphism classes of bi-orders and texts. Formally, bi-orders $\sigma = (\rho_1, \rho_2)$ and $\sigma' = (\rho'_1, \rho'_2)$ are *isomorphic* if there is a bijection $\psi : dom(\sigma) \to dom(\sigma')$ such that $\psi(\rho_1) = \rho'_1$ and $\psi(\rho_2) = \rho'_2$. The *length of* an (abstract) bi-order σ, denoted by $|\sigma|$, is $\#dom(\sigma')$ for some representative σ' of σ.

Texts $\tau = (\lambda, \rho_1, \rho_2)$ and $\tau' = (\lambda', \rho'_1, \rho'_2)$ are *isomorphic* if (ρ_1, ρ_2) and (ρ'_1, ρ'_2) are isomorphic and for the corresponding bijection $\psi : dom(\tau) \to dom(\tau')$, $\lambda' = \lambda \circ \psi^{-1}$. Hence isomorphic texts have the same word; this allows us to say that an (abstract) text is a pair (w, σ) where σ is an (abstract) bi-order, and w is a word of length $|\sigma|$. The *length of* $\tau = (w, \sigma)$, denoted by $|\tau|$, is $|\sigma|$.

Note that isomorphism of texts or bi-orders respects clans and hence also the above defined properties based on clans.

Sometimes, e.g., in examples, we have to give an abstract bi-order a concrete representation. We then write for a bi-order σ of length n simply the order $(i_1, \ldots, i_n)$ which comes from the representative $((1, 2, \ldots, n), (i_1, \ldots, i_n))$ with domain $\{1, 2, \ldots, n\}$. Accordingly, $\tau = (w, \sigma)$ is written as $(w, (i_1, \ldots, i_n))$. This notation is called the *standard form* of a bi-order (or of a text).

Note that the only bi-orders that are both primitive and sequential are the forward sequential bi-order of length 2, which will be denoted by σ_f in the sequel, the backward sequential bi-order of length 2, denoted by σ_b, and the bi-order of length 1. The set of

primitive bi-orders of length > 1 is denoted by PRIM. The bi-orders in PRIM$-\{\sigma_f, \sigma_b\}$ are called *strictly primitive*. There are infinitely many strictly primitive texts.

A text of length 1 is called a *singleton text*. A singleton text τ represented by $(\lambda, (x), (x))$ with $\lambda(x) = a \in \Delta$ is shortly denoted by $\underline{a}$.

4.2 Hierarchical representation of texts

In the theory of 2-structures, the notion of a clan underlies the decomposition of 2-structures by forming quotients. By repeatedly applying this quotient decomposition one obtains a decomposition tree which is a hierarchical representation of a 2-structure.

In the case of bi-orders, *bi-ordered trees* serve as hierarchical representations. This notion generalizes an ordered tree in that it is a tree t together with *two* orderings $ord_1(v)$ and $ord_2(v)$ associated to its inner nodes such that for each inner node v, $(ord_1(v), ord_2(v))$ is a bi-order on the children of v.

Remark 4.2 Note that equivalently, one can imagine a bi-ordered tree as an ordered tree where each inner node is labeled by an (abstract) bi-order. Then for an inner node v, its bi-order should be matched with its children in such a way that the first order is precisely the order $ord(v)$ from the tree. □

Given a bi-ordered tree t, t represents a bi-order as follows. Similar to the situation for ordered trees, the local linear orders $ord_1(v)$ and $ord_2(v)$ each induce a linear order on the leaves of t. The bi-order represented by t is (ρ_1, ρ_2), where ρ_1 and ρ_2 are the respective induced leave orderings.

Just as an ordered tree which is leaf-labeled hierarchically represents a word, viz. its yield, a leaf-labeled bi-ordered tree t represents the text (w, σ), where w is the yield of the leaf-labeled ordered tree obtained from t by forgetting the second ordering function ord_2, and σ is the bi-order represented by the underlying bi-ordered tree. The text represented by a leaf-labeled bi-ordered tree t is denoted by $txt(t)$.

Thus, a leaf-labeled bi-ordered tree is a hierarchical representation of a text. It corresponds with a decomposition of the text by repeatedly forming quotients into clans (analogous to the decomposition theory of 2-structures). Conversely, given a text τ, each decomposition tree of τ, obtained by repeatedly dividing into clans, is a leaf-labeled bi-ordered tree which represents τ as described above. All this is best illustrated by an example.

Example 4.3 Consider the leaf-labeled bi-ordered tree t from Figure 1.

For each inner node, the first associated order on the children is the left-to-right order, and the given label determines the second order (cf. Remark 4.2 — this label

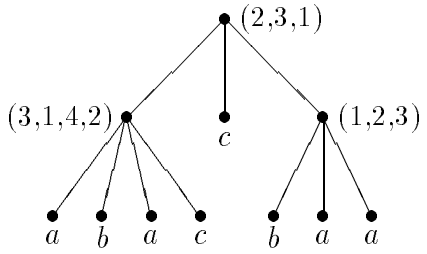


Figure 1: bi-ordered tree representing $\tau = (abaccbaa, (5, 6, 7, 8, 3, 1, 4, 2))$

is the standard form of the bi-order associated with the node). E.g., the second order on the children of the root is first the middle child, then the rightmost child, and then the leftmost child. By naming the leaves 1 to 8 from left to right, and reading from the tree the order on the leaves induced by the second orders, we obtain the standard form of $txt(t) : \tau = (abaccbaa, (5,6,7,8,3,1,4,2))$.

The tree t corresponds with the decomposition of τ into $\{1,2,3,4\}$, $\{5\}$, $\{6,7,8\}$ at the root level, followed by decomposing into singletons. Note that these subsets are indeed clans of τ, i.e., segments in both $(\overbrace{1,2,3,4}, \overbrace{5}, \overbrace{6,7,8})$ and $(\overbrace{5}, \overbrace{6,7,8}, \overbrace{3,1,4,2})$.

The node corresponding with $\{6,7,8\}$ can be further refined into $\{6\}$ and $\{7,8\}$, and thus we obtain the decomposition tree t_1, depicted as the leftmost tree in Figure 2. By additionally refining the root of the tree we obtain t_2, the middle tree in the figure. Note that the node with associated bi-order $(3,1,4,2)$ allows no further refinement. The rightmost tree t_3 gives another decomposition of τ, but t_3 is not obtained by refining t. Note that t_1, t_2, t_3 are indeed representing the text τ. □

Obviously, adding nodes with a single outgoing edge (i.e., chains) to a leaf-labeled bi-ordered tree does not change the represented text. Throughout this paper, bi-ordered trees are assumed not to have chains, unless they serve as "derivation trees" (see subsection 4.3).

For a (leaf-labeled) bi-ordered tree we write simply that an inner node *is* primitive (or sequential) if the node is labeled by a primitive (or sequential) bi-order.

A *primitive representation* of a text τ is a leaf-labeled bi-ordered tree representing τ such that each inner node is primitive. A primitive representation of a text τ corresponds with a "maximal" decomposition of τ in the sense that further decomposing is impossible. In general a text may have more than one primitive representation.

The *shape* of a text τ, denoted by $shape(\tau)$, is the unique leaf-labeled bi-ordered tree representing τ such that each inner node is primitive or sequential and, for each forward (backward) sequential node, none of its children is forward (backward, respectively) sequential. The notion of a shape comes from the theory of 2-structures. The shape is obtained by repeatedly decomposing into clans of maximal size that do not overlap other clans. This way of partitioning forces the quotients to be primitive or sequential;

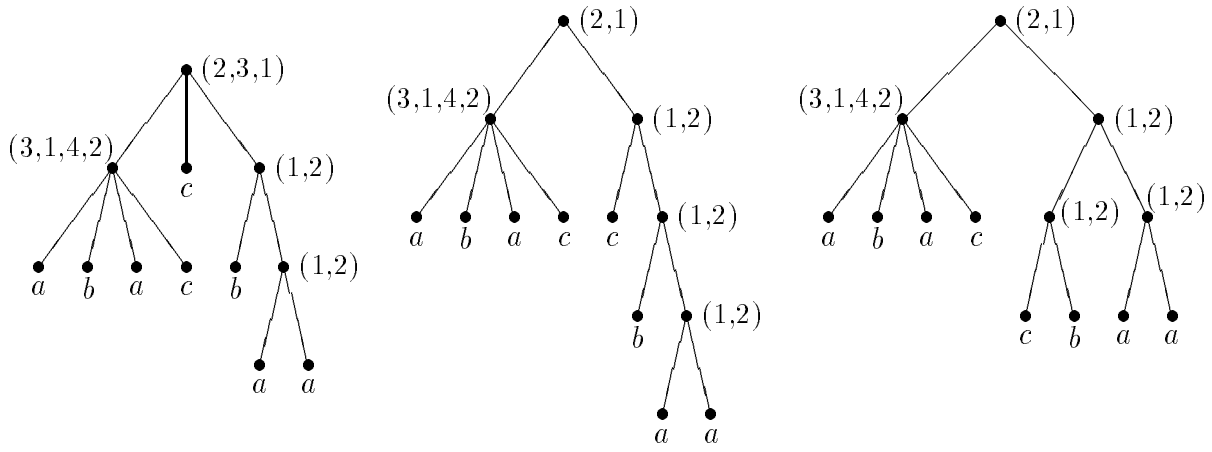


Figure 2: three more representations for $\tau = (abaccbaa, (5,6,7,8,3,1,4,2))$

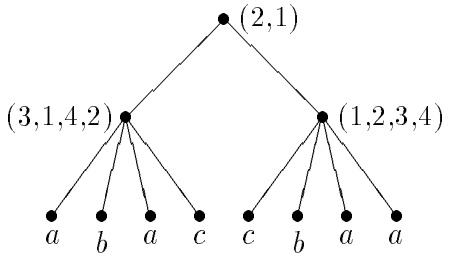


Figure 3: the shape of $\tau = (abaccbaa, (5,6,7,8,3,1,4,2))$

the uniqueness of the shape follows from the fact that such partitions are uniquely determined for each bi-order.

Example 4.4 In Example 4.3 neither t nor t_1 is a primitive representation or the shape of τ, since the root is neither primitive nor sequential. t_2 and t_3 are both primitive representations of the text τ. Neither of them is the shape of τ, since the second child of the root is forward sequential, and has a child which is also forward sequential. The shape of τ is given in Figure 3; its root is backward sequential, the second child of the root is forward sequential. □

Given a primitive representation of τ, the shape of τ can be obtained by contracting subsequent nodes with label σ_f into one forward sequential node, and contracting similarly nodes with label σ_b. This entails the following result, shown in [9] (see also [8]).

Proposition 4.5 *Each primitive representation of a text τ can be obtained from shape(τ) by refining the sequential nodes into subtrees the nodes of which have associated bi-orders σ_f or σ_b.* □

We denote by $op(\tau)$ the set of bi-orders occurring in a primitive representation of τ. By the above proposition this set is well defined.

A leaf-labeled bi-ordered tree t representing a text τ can be obtained by step-wise decomposing τ. Conversely, one can view the recovering of τ from t as a step-wise composition of τ. Then one step amounts to applying the operation of *simultaneous substitution*. For a bi-order σ of length $m \geq 1$, and texts $\tau_1 = (w_1, \sigma_1), \ldots, \tau_m = (w_m, \sigma_m)$, the text $[\sigma \leftarrow (\tau_1, \ldots, \tau_m)]$ is defined as follows : let $(\rho_1, \rho_2), (\rho_1^{(1)}, \rho_2^{(1)}), \ldots, (\rho_1^{(m)}, \rho_2^{(m)})$ be representatives of $\sigma, \sigma_1, \ldots \sigma_m$ with mutually disjoint domains, then $[\sigma \leftarrow (\tau_1, \ldots, \tau_m)]$ is the text $(w_1 \cdots w_m, \sigma_0)$ where σ_0 is the bi-order with representative $(\rho_1^{(\rho_1(1))} + \rho_1^{(\rho_1(2))} + \cdots + \rho_1^{(\rho_1(m))},\ \rho_2^{(\rho_2(1))} + \rho_2^{(\rho_2(2))} + \cdots + \rho_2^{(\rho_2(m))})$.

For a leaf-labeled bi-ordered tree t, $txt(t)$ can be obtained from t by repeatedly substituting texts corresponding with subtrees into the bi-order associated with the parent of these subtrees. More precisely, if t is a leaf-labeled bi-ordered tree where the root has associated bi-order σ, and the direct subtrees of the root are $t_1, \ldots, t_m$, then $txt(t) = [\sigma \leftarrow (txt(t_1), \ldots, txt(t_m))]$.

The following proposition gives a reformulation of the fact that for each text one can construct a primitive representation (see also [8]), and a consequence of the fact that such a primitive representation is a refinement of the shape (Proposition 4.5).

Proposition 4.6
(1) *Each text can be obtained from singleton texts by repeated substitution into primitive bi-orders.*
(2) *If* $\tau = [\sigma \leftarrow (\tau_1, \ldots, \tau_m)]$ *with* $\sigma \in \mathrm{PRIM}$*, then for each primitive representation of* τ *the root has bi-order* σ*; if* σ *is strictly primitive, then the direct subtrees of its root are primitive representations of* $\tau_1, \ldots, \tau_m$*, respectively.* □

We will also use the notion of singular substitution of texts, which is a special case of simultaneous substitution. For a text τ of length m with $word(\tau) = a_1 \cdots a_m$, a text τ', and $1 \leq i \leq m$, the *substitution of* τ' *into* τ *at* i, denoted by $subst(\tau, i, \tau')$, is the text $[\sigma \leftarrow (\underline{a_1}, \ldots, \underline{a_{i-1}}, \tau', \underline{a_{i+1}}, \ldots, \underline{a_m})]$. (Recall that here $\underline{a_j}$ denotes the singleton text with word a_j). Singular substitution underlies the notion of a derivation step in a text grammar.

4.3 Text grammars

A set of texts K is called a *text language*. For an alphabet Δ, $\mathrm{TXT}(\Delta)$ denotes the set of all texts over Δ. Also, for a finite subset Π of PRIM, we use $\mathrm{TXT}_\Pi(\Delta)$ to denote the set of texts over Δ that have a primitive representation using only bi-orders from Π, i.e., $\mathrm{TXT}_\Pi(\Delta)$ is the set of texts τ over Δ for which $op(\tau) \subseteq \Pi$.

A *context-free text grammar* is a 4-tuple $G = (N, \Delta, P, \tau_0)$, where N and Δ are disjoint alphabets, P is a finite set of productions $A \to \tau$, where $A \in N$ and $\tau \in \mathrm{TXT}(N \cup \Delta)$, and τ_0 is a singleton text over N. As usual, we say that the letters of Δ are the terminals, and that the letters of N are the nonterminals of the grammar.

Let $G = (N, \Delta, P, \tau_0)$ be a text grammar.

Let $\tau = (w, \sigma)$ and τ' be texts in $\mathrm{TXT}(N \cup \Delta)$. τ (directly) *derives* τ' *(in* G*)*, denoted $\tau \Rightarrow_G \tau'$, if there is a production $A \to \nu \in P$ and $1 \leq i \leq |\tau|$ with $w(i) = A$ such that $\tau' = subst(\tau, i, \nu)$.

The transitive closure of $\Rightarrow_G$ is denoted by $\Rightarrow_G^+$, and the reflexive and transitive closure by $\Rightarrow_G^*$. We omit the subscript G whenever the grammar G is clear from the context.

$\mathrm{TxL}(G)$ denotes the *text language generated by* G, i.e., $\mathrm{TxL}(G) = \{\tau \in \mathrm{TXT}(\Delta) \mid \tau_0 \Rightarrow_G^* \tau\}$.

Next we define derivation trees in the text grammar G. First recall that in a bi-ordered tree, with each inner node v a bi-order $(ord_1(v), ord_2(v))$ on its children is associated. Now in a *node-labeled* bi-ordered tree, we can associate with each inner node v a *text* on its children, where the word of the text is formed by the labels of the children according to the first ordering $ord_1(v)$. For a node-labeled bi-ordered tree t, we denote by $di(t)$ the leaf-labeled bi-ordered tree that is obtained from t by removing the labels of the inner nodes, and, if occurring, its chains.

Now for a text $\tau \in \mathrm{TXT}(N \cup \Delta)$ and $A \in N$, a *derivation tree of* τ *from* A in G is a node-labeled bi-ordered tree t such that the root has label A, $di(t)$ represents τ, and for each inner node v of t, the production $B \to \nu$ is a production of P, where B is the label of v and ν is the text associated to v as described above. As usual, for $A \in N$, and $\tau \in \mathrm{TXT}(N \cup \Delta)$, $\underline{A} \Rightarrow^* \tau$ iff there is a derivation tree of τ from A. By a *derivation tree* we mean a derivation tree of some $\tau \in \mathrm{TXT}(\Delta)$ from S, where S is the nonterminal specifying τ_0. We denote by $\mathrm{Di}(G)$ the set $\{di(t) \mid t$ is a derivation tree in $G\}$.

A text language K is a *context-free text language* if there exists a context-free text grammar G such that $K = \mathrm{TxL}(G)$. In [9] it was shown that every context-free text language has finitely many primitive subtexts. Consequently, we have the following result, where $op(K) = \{op(\tau) \mid \tau \in K\}$.

Proposition 4.7 *For each context-free text language K, $op(K)$ is finite.* □

Hence for every context-free text language K, there exists a finite subset $\Pi \subset \mathrm{PRIM}$ and an alphabet Δ such that $K \subseteq \mathrm{TXT}_\Pi(\Delta)$.

By a standard construction productions of the form $A \to \underline{B}$ with A, B nonterminals can be eliminated, i.e., each context-free text grammar has an equivalent chain-free grammar. Therefore, in what follows we assume that each context-free text grammar is chain-free. Hence the only possible chains in derivation trees are ending in a leaf. Also, obviously, we may assume that text grammars are reduced, i.e., for each nonterminal A there is a derivation of a text of the generated language that uses A.

By decomposing the right-hand sides one obtains for each context-free grammar an equivalent text grammar in so-called primitive normal form. A context-free text grammar $G = (N, \Delta, P, \tau_0)$ is in *primitive normal form*, abbreviated PNF, if for each production $A \to \tau$ in P, τ is a primitive text. G is in *Chomsky-like primitive normal form*, abbreviated CPNF, if G is in PNF and for each production $A \to \tau$, either $word(\tau) \in \Delta$, or $word(\tau) \in N^+$.

Note that for each derivation tree t of a context-free text grammar in PNF $di(t)$ is a primitive representation of the generated text. For the following class of text grammars the derivation trees are the *shapes* of the generated texts.

Definition 4.8
(1) A context-free text grammar G is *shapely* if $\mathrm{Di}(G) = \{shape(\tau) \mid \tau \in \mathrm{TxL}(G)\}$.
(2) A text language K is *shapely* if there is a shapely grammar generating K.

A text language K is *limited* iff there exists a constant C such that, for each $\tau \in K$, the outdegree of the nodes in the shape of τ is bounded by C. By Proposition 4.7, for a context-free text language the outdegree of primitive nodes in the shapes is bounded. However, the sequential nodes in the shapes may in general be of unbounded outdegree. The requirement that these too are bounded forms a necessary and also sufficient condition for the language to be shapely.

Proposition 4.9 ([9, Theorem 6.1]) *A text language K is shapely iff K is context-free and limited.* □

To show the non-context-freeness of text languages, we may use the following pumping lemma, where the meaning of the notation $subst^k(\tau, i, \tau')$ is inductively defined as τ' if $k = 0$, and as $subst(\tau, i, subst^{k-1}(\tau, i, \tau'))$ for $k > 0$.

Proposition 4.10 ([9, Theorem 7.3]) *Let K be a context-free text language. There exist constants p and q such that for each $\tau \in K$ with $|\tau| > p$, there exist texts τ_1, τ_2, τ_3, $1 \leq i \leq |\tau_1|$, $1 \leq j \leq |\tau_2|$ such that*
(1) $\tau = subst(\tau_1, i, subst(\tau_2, j, \tau_3))$,
(2) $|\tau_2| > 1$,
(3) $|subst(\tau_2, j, \tau_3)| \leq q$,
(4) *for each $k \geq 0$, $subst(\tau_1, i, subst^k(\tau_2, j, \tau_3)) \in K$.* □

5 An algebra of texts

We give an algebraic structure to the set of texts $\mathrm{TXT}_\Pi(\Delta)$, where Π is a finite subset of PRIM, and Δ is a (finite) alphabet. Each of the primitive bi-orders $\pi \in \Pi$ will act as an operator on $\mathrm{TXT}_\Pi(\Delta)$; its associated mapping is the simultaneous substitution in π.

Let $\Sigma = \Pi \cup \Delta$ be the ranked alphabet such that the rank of each $\sigma \in \Pi$ is $|\sigma|$ and the rank of each $a \in \Delta$ is 0. Then $\mathcal{T}_\Sigma = (\mathrm{TXT}_\Pi(\Delta), \Sigma)$ is the Σ-algebra defined by $\sigma^{\mathcal{T}_\Sigma}(\tau_1, \ldots, \tau_m) = [\sigma \leftarrow (\tau_1, \ldots, \tau_m)]$ for $\sigma \in \Sigma_m$, $m \geq 2$, $\tau_1, \ldots, \tau_m \in \mathrm{TXT}_\Pi(\Delta)$, and $a^{\mathcal{T}_\Sigma} = \underline{a}$ for $a \in \Sigma_0$.

Let $\mathcal{F}_\Sigma = (F_\Sigma, \Sigma)$ be the term Σ-algebra. As mentioned in subsection 3.1 we think of the elements of F_Σ as trees; for this specific choice of Σ as $\Pi \cup \Delta$, the trees in F_Σ are, by Remark 4.2, primitive bi-ordered trees which are hierarchical representations of texts over Δ. Hence the notion of a primitive representation of a text is in this setting an algebraic expression of a text.

The mapping $txt : F_\Sigma \to \mathrm{TXT}_\Pi(\Delta)$ which assigns to each $t \in F_\Sigma$ the text $txt(t)$ represented by t is a homomorphism of Σ-algebras. First of all, it should be noted that $txt(F_\Sigma) \subseteq \mathrm{TXT}_\Pi(\Delta)$, because, for each tree $t \in F_\Sigma$, $op(txt(t))$ consists of bi-orders labeling inner nodes of t, and so $op(txt(t)) \subseteq \Pi$. Also, txt is indeed a homomorphism, since for each $a \in \Delta$, $txt(a) = \underline{a} = a^{\mathcal{T}_\Sigma}$, and for each $\sigma \in \Sigma_m$, $m \geq 2$, and all $t_1, \ldots, t_m \in F_\Sigma$, $txt(\sigma\langle t_1 \cdots t_m \rangle) = [\sigma \leftarrow (txt(t_1), \ldots, txt(t_m))]$, as explained in subsection 4.2.

Hence txt is the unique homomorphism from the initial term Σ-algebra $\mathcal{F}_\Sigma$ to $\mathcal{T}_\Sigma$. By Proposition 4.6(1), txt is surjective. We conclude that $\mathcal{T}_\Sigma$ is generated by Σ and that $\mathcal{T}_\Sigma$ is isomorphic with the quotient algebra $\mathcal{F}_\Sigma / \ker(txt)$.

Consider the congruence given by the kernel of txt. Two primitive bi-ordered trees are in the same congruence class iff they represent the same text. By Proposition 4.5, such trees differ only in the way the sequential nodes of the shape are refined. It follows that $\ker(txt)$ is precisely the congruence generated by the equations $\sigma_f\langle u \sigma_f \langle vw \rangle\rangle = \sigma_f\langle \sigma_f \langle uv \rangle w\rangle$, $\sigma_b\langle u \sigma_b \langle vw \rangle\rangle = \sigma_b\langle \sigma_b \langle uv \rangle w\rangle$, where u, v, w are variables. Hence $\mathcal{T}_\Sigma$ is a T-algebra, where T is the theory (Σ, E) such that E is the set consisting of the above two equations expressing the associativity of σ_f and σ_b. Moreover, $\mathcal{T}_\Sigma$, being isomorphic with the quotient term algebra $\mathcal{F}_\Sigma / \ker(txt)$, is initial in the class of T-algebras. Of course, if some of the operations σ_f, σ_b are not in Σ, then we restrict E to a subset of these equations. In particular, if $\Sigma \cap \{\sigma_f, \sigma_b\} = \emptyset$, then $E = \emptyset$, and $\mathcal{T}_\Sigma$ is isomorphic with the term Σ-algebra $\mathcal{F}_\Sigma$.

Note also that if $\Pi = \{\sigma_x\}$, with $x \in \{f, b\}$, then the corresponding Σ-algebra of sequential texts $\mathcal{T}_\Sigma = (\mathrm{TXT}_{\{\sigma_x\}}(\Delta), \Sigma)$, with $\Sigma = \{\sigma_x\} \cup \Delta$, is isomorphic with the semi-group Δ^+ seen as a Σ-algebra (see subsection 3.1).

Remark 5.1
One could add the (primitive) bi-order of length 1 as a unary operation to the ranked alphabet Σ. Its interpretation in $\mathcal{T}_\Sigma$ is the identity, and the terms in F_Σ describe then also trees with chains. If the equation '$\sigma_1(v) = v$', where σ_1 stands for the bi-order of length 1 is added to E, then again $\mathcal{T}_\Sigma$ is a T-algebra, with $T = (\Sigma, E)$. Including σ_1 in this way would not affect any of the results in this paper, but for technical simplicity we have chosen to leave it out. □

We consider recognizability and equationality of text languages, interpreting Definitions 3.1 and 3.3 in a Σ-algebra $\mathcal{T}_\Sigma$ of texts as described above. Note that if K is a

recognizable (or equational) text language in this sense, then for every choice of Π and Δ such that $K \subseteq \mathrm{TXT}_\Pi(\Delta)$, K is recognizable (or equational) w.r.t. the corresponding ranked alphabet $\Sigma = \Pi \cup \Delta$.

In particular, for a forward sequential text language K, K is recognizable or equational iff K is a recognizable or equational subset of $\mathrm{TXT}_{\{\sigma_f\}}(\Delta)$ iff the underlying word language is recognizable or equational w.r.t. the isomorphic $\{\sigma_f\} \cup \Delta$-algebra Δ^+. This immediately provides easy examples of text languages that are equational but not recognizable, e.g., the text language $\{(a^n b^n, (1, \ldots, 2n)) \mid n \geq 1\}$.

We will show that the equational languages are precisely the context-free text languages. Recall that every context-free language, as every equational language, is a subset of $\mathrm{TXT}_\Pi(\Delta)$ for some finite $\Pi \subset \mathrm{PRIM}$ and Δ.

Remark 5.2 There is a close correspondence between context-free text grammars for text languages in $\mathrm{TXT}_\Pi(\Delta)$, and regular tree grammars generating tree languages in F_Σ.

For a regular tree grammar $H = (N, \Delta, P, S)$, we denote by $txt(H)$ the text grammar $(N, \Delta, P', \underline{S})$, where $P' = \{A \to txt(t) \mid A \to t \in P\}$. Then $\mathrm{TxL}(txt(H)) = txt(\mathrm{TrL}(H))$. If H is in normal form, then $txt(H)$ is in CPNF, and $\mathrm{Di}(txt(H)) = \mathrm{TrL}(H)$.

Conversely, let $G = (N, \Delta, P, \underline{S})$ be a context-free text grammar. Let $H = (N, \Delta, P', S)$ be a regular tree grammar such that P' contains for each $A \to \tau \in P$ one production $A \to t$ where $t \in F_\Sigma$ is such that $txt(t) = \tau$. Then $txt(\mathrm{TrL}(H)) = \mathrm{TxL}(G)$. If G is in CPNF, then H is in normal form, and $\mathrm{Di}(G) = \mathrm{TrL}(H)$. □

Example 5.3 Let $G = (N, \Delta, P, \underline{S})$ be the context-free text grammar such that $N = \{S, A, C\}, \Delta = \{a, c\}$, and P consists of the productions
$S \to (AS, (1,2)), A \to (aC, (2,1)), C \to (cCac, (3,1,4,2)), S \to \underline{a}, C \to \underline{c}, A \to \underline{a}$.

Then the regular tree grammar $H = (N, \Delta, P', S)$ generates $\mathrm{Di}(G)$, where P' consists of the productions $S \to \sigma_f\langle AS\rangle$, $A \to \sigma_b\langle aC\rangle$, $C \to \pi\langle cCac\rangle$, $S \to a$, $C \to c$, $A \to a$, where π is the abstract bi-order with standard form $(3,1,4,2)$. □

Lemma 5.4 *A text language is equational iff it is context-free.*

Proof. Let K be a text language. It follows from Remark 5.2 that K is context-free iff there exists a tree language T, generated by a regular tree grammar, such that $txt(T) = K$. By Theorem 3.4(1) and (5) and the fact that a tree language is generated by a regular tree grammar iff it is equational (see subsection 3.1), it follows that K is context-free iff K is equational. □

Through this connection between polynomial systems and text grammars, the construction in [12, Lemma 3.1] which yields a "normal form" for polynomial systems is related to the result (in [9]) that each context-free text language has a context-free text grammar in CPNF. Also, this connection is a special case of the situation described in [4], see also [3], where it is shown that given a Σ-algebra $\mathcal{A} = (A, \Sigma)$, one can define a well-behaved substitution device in A such that the equational subsets of A given by a polynomial system are precisely the sets generated (using this substitution) by an "abstract" context-free grammar.

We end this section by formulating the consequences for text languages following from subsection 3.1 and this section.

Theorem 5.5 *Let $K \subseteq \mathrm{TXT}_\Pi(\Delta)$ be a text language.*
(1) K is context-free iff it equals $txt(T)$ for some recognizable tree language T.
(2) K is recognizable iff $txt^{-1}(K)$ is a recognizable tree language.
(3) If K is recognizable, then K is context-free. □

6 Recognizable text languages

We have seen that equational text languages coincide with the text languages generated by context-free text grammars, which were investigated in [9].

We now consider the family of recognizable text languages. Like for generated algebras in general, in the case of texts the class of recognizable sets is included in the class of equational sets. Hence, each recognizable text language is generated by a context-free text grammar (Theorem 5.5(3)). Like for words, but unlike trees, this inclusion of recognizable sets in equational sets is strict. As the main result of this section we give a grammatical characterization of the recognizable text languages by restricting the context-free text grammars to a natural subclass. This generalizes to texts the well-known characterization of regular word languages by right-linear grammars.

Previously we have defined recognizability of text languages using finite algebras. Reformulating Proposition 3.2, which characterizes the recognizable subsets of an algebra in terms of their Nérode congruences, we obtain the following result.

Lemma 6.1 *A text language $K \subseteq \mathrm{TXT}_\Pi(\Delta)$ is recognizable iff the congruence $\cong_K$ is finite, where for $\tau_1, \tau_2 \in \mathrm{TXT}_\Pi(\Delta)$, $\tau_1 \cong_K \tau_2$ iff for all $\tau \in \mathrm{TXT}_\Pi(\Delta)$, and for all i with $1 \leq i \leq |\tau|$, $subst(\tau, i, \tau_1) \in K$ iff $subst(\tau, i, \tau_2) \in K$.* □

Each text has a natural structure, its shape. As we have discussed, all primitive hierarchical representations of a text differ from the shape only by the refinement of sequential nodes into binary subtrees. This implies that if the root of the shape is strictly primitive, then this is the root of every primitive representation of the text (see Proposition 4.6(2)). As a derivation tree for a context-free text grammar gives a representation for the derived text we can translate this observation to derivations in text grammars.

Lemma 6.2 *Let $G = (N, \Delta, P, \tau_0)$ be a context-free text grammar in PNF. Let $\sigma \in$ PRIM with $|\sigma| = m > 2$, and let $\tau = [\sigma \leftarrow (\tau_1, \ldots, \tau_m)]$, where $\tau_1, \ldots, \tau_m \in \mathrm{TXT}(N \cup \Delta)$. Then $\underline{A} \Rightarrow^* \tau$ iff there exist $A_1, \ldots, A_m \in N \cup \Delta$ such that $A \to (A_1 \cdots A_m, \sigma)$ and $\underline{A_j} \Rightarrow^* \tau_j$ for $j = 1, \ldots, m$.* □

More generally, in a derivation tree of a text the subtrees consisting of strictly primitive nodes are determined by the text itself (i.e., by its shape). The only structural freedom in deriving a text lies in the possible decompositions for the sequential nodes of the shape. A natural restriction to context-free text grammars is to force the grammar to choose right-linear derivations for these sequential (word-like) substructures. In other words, we forbid left-recursion in derivation trees: subtrees of the form $\sigma_x\langle\sigma_x\langle t_1 t_2\rangle t_3\rangle$ where $x \in \{f, b\}$. Note that each text τ has a unique primitive representation that has no left recursion; we denote this tree by $nlr(\tau)$.

We formulate this requirement in terms of productions, rather than in terms of derivation trees.

Definition 6.3 A context-free text grammar $G = (N, \Delta, P, \tau_0)$ is *right-linear* if G is in PNF and for each production $A \to (BC, \sigma_x) \in P$, with $A, B \in N, C \in N \cup \Delta$, and $x \in \{b, f\}$, if $B \to (w, \sigma) \in P$, then $\sigma \neq \sigma_x$.

Example 6.4 Let $G = (N, \Delta, P, \underline{S})$ be the context-free text grammar such that $N = \{S, A, B, C\}, \Delta = \{a, b, c\}$, and P consists of the productions $S \to (AS, (1,2))$, $S \to \underline{b}$, $A \to (aB, (2,1))$, $B \to (AB, (1,2))$, $B \to \underline{b}$, $A \to (cB, (1,2))$. Then G is not right-linear, since $S \to (AS, (1,2)) \in P$, and $A \to (cB, (1,2)) \in P$. □

Lemma 6.5 *A context-free text grammar G is right-linear iff* $\mathrm{Di}(G) = nlr(\mathrm{TxL}(G))$. □

The following observation turns out to be crucial in our considerations on right-linear grammars. It is a reformulation of the intuition that sequential substructures of a text are generated by the grammar in a right-linear way. We say that a text τ is *of type* $x \in \{f, b\}$ if the root of a primitive representation of τ has label σ_x (cf. Proposition 4.6(2)).

Lemma 6.6 *Let $G = (N, \Delta, P, \tau_0)$ be a right-linear text grammar.*
$\underline{A} \Rightarrow^* [\sigma_x \leftarrow (\tau_1, \tau_2)]$ for $x \in \{f, b\}$, $A \in N$, and texts τ_1 and τ_2 over $N \cup \Delta$, iff there exists a $B \in N$ such that $\underline{A} \Rightarrow^* [\sigma_x \leftarrow (\tau_1, \underline{B})]$ and $\underline{B} \Rightarrow^* \tau_2$. Additionally, if τ_1 is not of type x, then $\underline{A} \Rightarrow^* [\sigma_x \leftarrow (\tau_1, \tau_2)]$ iff there exists a production $A \to (CB, \sigma_x)$, where $C \in N$ is such that $\underline{C} \Rightarrow^* \tau_1$. □

It is perhaps instructive to notice that in the case of words this lemma says that in a right-linear grammar $A \Rightarrow^* w_1 w_2$ iff there is a B such that $A \Rightarrow^* w_1 B$ and $B \Rightarrow^* w_2$.

In the definition of right-linearity we have forced the context-free grammar to generate texts according to a specific structure on the derivation trees. We will now define a dual class of grammars. Rather than choosing one normal form for the derivation trees we will impose on the grammar that if it generates a text in any way, then it can also do so according to all primitive representations of the text. This notion generalizes the property of Lemma 6.2 to the case where $|\sigma| = 2$.

Definition 6.7 A context-free text grammar $G = (N, \Delta, P, \tau_0)$ is *complete* if G is in PNF and for each $A \in N$, and for each $\tau = [\sigma_x \leftarrow (\tau_1, \tau_2)]$, where $x \in \{f, b\}$ and $\tau_1, \tau_2 \in \mathrm{TXT}(\Delta)$, if $\underline{A} \Rightarrow^* \tau$, then there exist $A_1, A_2 \in N \cup \Delta$ such that $A \to (A_1 A_2, \sigma_x) \in P$ and $\underline{A_j} \Rightarrow^* \tau_j$ for $j = 1, 2$.

Note that every complete text grammar has an equivalent complete text grammar in CPNF.

The completeness property is perhaps more intuitive when stated in terms of derivation trees of the grammar.

Lemma 6.8 *A context-free text grammar G is complete iff* $\mathrm{Di}(G) = txt^{-1}(\mathrm{TxL}(G))$. □

It turns out that complete grammars and right-linear grammars characterize recognizable text languages.

Theorem 6.9 *Let K be a text language. The following statements are equivalent.*
(1) *K is recognizable.*
(2) *There is a complete context-free text grammar G such that $K = \mathrm{TxL}(G)$.*
(3) *There is a right-linear context-free text grammar G such that $K = \mathrm{TxL}(G)$.*

Proof. Note that (in each of the three cases) we can assume that Π and Δ are given such that $K \subseteq \mathrm{TXT}_\Pi(\Delta)$; let $\Sigma = \Pi \cup \Delta$ be the corresponding ranked alphabet.

(1) $\Rightarrow$ (2). If K is recognizable, then by Theorem 2.4(2), $txt^{-1}(K) \subseteq F_\Sigma$ is recognizable, and hence there is a regular tree grammar H for $txt^{-1}(K)$. We may assume that H is in normal form. Since $K = txt(txt^{-1}(K))$, the context-free text grammar $G = txt(H)$ generates K, and $\mathrm{Di}(G) = txt^{-1}(K)$ (see Remark 2.1). Hence, by Lemma 6.8, G is complete.

(2) $\Rightarrow$ (3). Let $G = (N, \Delta, P, \tau_0)$ be a complete context-free text grammar in CPNF such that $\mathrm{TxL}(G) = K$. We transform G into a right-linear text grammar by forcing it to choose right-linear derivation trees.

Formally, let $G' = (N', \Delta, P', \tau_0)$ be the context-free text grammar with

$$\begin{aligned} N' &= N \cup \{A^f, A^b \mid A \in N\}, \text{ and} \\ P' &= \{A' \to (w, \sigma) \mid A \to (w, \sigma) \in P, A' \in \{A, A^f, A^b\}, \sigma \notin \{\sigma_b, \sigma_f\}\} \\ &\cup \{A' \to (B^x C, \sigma_x) \mid A \to (BC, \sigma_x) \in P, A' \in \{A, A^y\} \text{ with } y \neq x, x \in \{f, b\}\}. \end{aligned}$$

It is not difficult to see that G' is a right-linear context-free text grammar, and that $\mathrm{TxL}(G') \subseteq \mathrm{TxL}(G)$. Since G is complete, by Lemma 6.8, for each text $\tau \in \mathrm{TxL}(G)$ there is a derivation tree in G without left recursion. This derivation tree can be made into a derivation tree of τ in G' by adding superscripts f and b to some of the non-terminal labels. Hence $\mathrm{TxL}(G) = \mathrm{TxL}(G')$.

(3) $\Rightarrow$ (1). Let $G = (N, \Delta, P, \underline{S})$ be a right-linear grammar in CPNF such that $\mathrm{TxL}(G) = K$. Based on G, we will define a finite Σ-algebra $\mathcal{Q} = (Q, \Sigma)$, and a homomorphism $h : \mathcal{T}_\Sigma \to \mathcal{Q}$ such that $K = h^{-1}(F)$ for some $F \subseteq Q$.

Let W be the set $N \times \{f, b\} \times N$. Let $Q = 2^{N \cup W}$, and let $\mathcal{Q} = (Q, \Sigma)$ be the Σ-algebra defined as follows :

(i) for $a \in \Sigma_0$, let $V = \{A \in N \mid A \to \underline{a} \in P\}$. Then

$$a^{\mathcal{Q}} = V \cup \{(A, x, C) \in W \mid A \to (BC, \sigma_x) \in P, B \in V\}$$

(ii) for $\sigma \in \Sigma_m$, $m > 2$, $V_1, \ldots, V_m \in Q$, let

$$V = \{A \in N \mid A \to (A_1 \cdots A_m, \sigma) \in P, A_i \in V_i \text{ for } i = 1, \ldots, m\}.$$

Then

$$\sigma^{\mathcal{Q}}(V_1, \ldots, V_m) = V \cup \{(A, x, C) \in W \mid A \to (BC, \sigma_x) \in P, B \in V\}$$

(iii) for $x \in \{f, b\}$, $V_1, V_2 \in Q$, let

$$V = \{A \in N \mid (A, x, C) \in V_1, C \in V_2\}.$$

Then

$$\begin{aligned} \sigma_x^{\mathcal{Q}}(V_1, V_2) &= V \\ &\cup \{(A, x, C) \in W \mid (A, x, B) \in V_1, (B, x, C) \in V_2\} \\ &\cup \{(A, y, C) \in W \mid A \to (BC, \sigma_y) \in P, B \in V\}, \text{ where } y \neq x. \end{aligned}$$

For notational convenience, we will use $\tau \oplus_x \underline{B}$ as shorthand for $[\sigma_x \leftarrow (\tau, \underline{B})]$, where $x \in \{f, b\}$, $B \in N$, and $\tau \in \mathrm{TXT}_\Pi(\Delta)$.

Let $h : \mathrm{TXT}_\Pi(\Delta) \to Q$ be the mapping such that

$$h(\tau) = \{A \in N \mid \underline{A} \Rightarrow^* \tau\} \cup \{(A, x, C) \in W \mid \underline{A} \Rightarrow^* \tau \oplus_x \underline{C}\}$$

Claim 6.10 *h is a homomorphism from $\mathcal{T}_\Sigma$ to $\mathcal{Q}$.*

Proof.

(i) Let $a \in \Delta$. Since G is in CPNF, $\underline{A} \Rightarrow^* \underline{a}$ iff $A \to \underline{a} \in P$, and $\underline{A} \Rightarrow^* \underline{a} \oplus_x \underline{C}$ iff there is a $B \in N$ such that $A \to (BC, \sigma_x) \in P$ and $B \to \underline{a} \in P$. It follows that $h(a^{\mathcal{T}_\Sigma}) = h(\underline{a}) = a^{\mathcal{Q}}$.

(ii) Let $\sigma \in \Sigma_m$, $m > 2$, let $\tau_1, \ldots, \tau_m \in \mathrm{TXT}_\Pi(\Delta)$, and let $\tau = \sigma^{\mathcal{T}_\Sigma}(\tau_1, \ldots, \tau_m) = [\sigma \leftarrow (\tau_1, \ldots, \tau_m)]$. By Lemma 6.2, and since G is in CPNF, $\underline{A} \Rightarrow^* \tau$ iff there exist $A_1, \ldots, A_m \in N$ such that $A \to (A_1 \cdots A_m, \sigma)$ and $\underline{A_j} \Rightarrow^* \tau_j$ for $j = 1, \ldots m$.

Using Lemma 6.6, we see that $\underline{A} \Rightarrow^* \tau \oplus_x \underline{C}$ iff there is a $B \in N$ such that $A \to (BC, \sigma_x) \in P$ and $\underline{B} \Rightarrow^* \tau$. Hence, $A \in h(\tau)$ iff there exists $A \to (A_1 \cdots A_m, \sigma) \in P$ such that $A_j \in h(\tau_j)$ for $j = 1, \ldots, m$, and $(A, x, C) \in h(\tau)$ iff there exists $A \to (BC, \sigma_x) \in P$ such that $B \in h(\tau)$.

Consequently $h(\tau) = h(\sigma^{\mathcal{T}_\Sigma}(\tau_1, \ldots, \tau_m)) = \sigma^{\mathcal{Q}}(h(\tau_1), \ldots, h(\tau_m))$.

(iii) Let $x \in \{f, b\}$, let $\tau_1, \tau_2 \in \mathrm{TXT}_\Pi(\Delta)$, and let $\tau = \sigma_x^{\mathcal{T}_\Sigma}(\tau_1, \tau_2) = [\sigma_x \leftarrow (\tau_1, \tau_2)]$. By Lemma 6.6, $\underline{A} \Rightarrow^* \tau$ iff there exists a $C \in N$ such that $\underline{A} \Rightarrow^* \tau_1 \oplus_x \underline{C}$ and $\underline{C} \Rightarrow^* \tau_2$. Hence $A \in h(\tau)$ iff $(A, x, C) \in h(\tau_1)$ and $C \in h(\tau_2)$ for some $C \in N$.

Since $\tau = \sigma_x^{\mathcal{T}_\Sigma}(\tau_1, \tau_2)$, we may write $\tau \oplus_x \underline{C} = [\sigma_x \leftarrow (\tau_1, \tau_2 \oplus_x \underline{C})]$ due to the associativity of σ_x. As before, $\underline{A} \Rightarrow^* \tau \oplus_x \underline{C}$ iff there exists a $B \in N$ such that $\underline{A} \Rightarrow^* \tau_1 \oplus_x \underline{B}$ and $\underline{B} \Rightarrow^* \tau_2 \oplus_x \underline{C}$. Hence $(A, x, C) \in h(\tau)$ iff $(A, x, B) \in h(\tau_1)$ and $(B, x, C) \in h(\tau_2)$ for some $B \in N$.

Using once more Lemma 6.6, observe that $\underline{A} \Rightarrow^* \tau \oplus_y \underline{C}$ iff there is a $B \in N$ such that $A \to (BC, \sigma_y) \in P$ and $\underline{B} \Rightarrow^* \tau$. Hence for $y \neq x$, $(A, y, C) \in h(\tau)$ iff $A \to (BC, \sigma_y) \in P$ and $B \in h(\tau)$ for some $B \in N$.

By combining the above three cases it follows that $h(\tau) = \sigma_x^{\mathcal{Q}}(h(\tau_1), h(\tau_2))$.

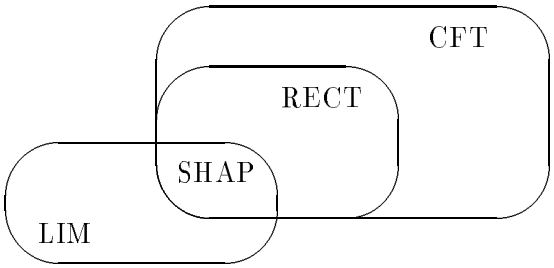
□

Note that the above homomorphism h from $\mathcal{T}_\Sigma$ to $\mathcal{Q}$ is unique, by the fact that $\mathcal{T}_\Sigma$ is an initial T-algebra over the theory T given in the previous section. Also $\mathcal{Q}$ itself is a T-algebra, since σ_f and σ_b are associative in $\mathcal{Q}$.

Now let $F \subseteq Q$ be the set $\{V \in Q \mid S \in V\}$. Then $h^{-1}(F) = \{\tau \in \mathrm{TXT}_\Pi(\Delta) \mid h(\tau) \in F\} = \{\tau \in \mathrm{TXT}_\Pi(\Delta) \mid S \in h(\tau)\} = \{\tau \in \mathrm{TXT}_\Pi(\Delta) \mid \underline{S} \Rightarrow^* \tau\} = K$. Hence K is recognizable. □

Theorem 6.9 can be understood as follows. Each context-free text language $\mathrm{TxL}(G)$ is of the form $txt(\mathrm{Di}(G))$, where $\mathrm{Di}(G)$ is a recognizable tree language. By requiring that G is complete, it is ensured that $\mathrm{Di}(G)$ is a so-called "saturated" subset of trees; right-linearity of G ensures that $\mathrm{Di}(G)$ is a subset of "well-formed representatives". For both types of recognizable tree languages we have that the corresponding text languages are recognizable.

The next theorem says that shapely text languages form a proper subclass of the class of recognizable text languages.



æ

Figure 4: families of text languages

Theorem 6.11 *Let K be a text language. The following statements are equivalent.*
(1) *K is shapely.*
(2) *K is recognizable and limited.*
(3) *K is context-free and limited.*

Proof.
(1) $\Leftrightarrow$ (3) This is Proposition 4.9.
(1) $\Rightarrow$ (2) Let G be a reduced shapely grammar for K. By Proposition 4.9, K is limited. We continue by showing that K is recognizable. G can be transformed into an equivalent text grammar by replacing each sequential production $A \to (B_1 \cdots B_m, \sigma)$ by the set of productions $A \to (B_1 A_1, \sigma_x)$, $A_1 \to (B_2 A_2, \sigma_x), \ldots, A_{m-2} \to (B_{m-1} B_m, \sigma_x)$, where $A_1, \ldots, A_{m-2}$ are new nonterminals, and $x = f$ or $x = b$ when σ is forward or backward sequential, respectively.
Note that from the shapeliness of G it follows that there are no sequential productions (of the same type as σ) applicable to any of the nonterminals $B_1, \ldots, B_m$. Hence the resulting grammar is right-linear, and, by Theorem 6.9, K is recognizable.
(2) $\Rightarrow$ (3) Follows immediately from Theorem 5.5(3). □

The diagram in Figure 4 represents the situation. Here CFT, RECT, SHAP, and LIM denote the families of all context-free, recognizable, shapely, and limited text languages, respectively.

7 Comparing text, word, tree languages

In comparing texts with words and trees, we take two approaches. First, more or less on the surface, one may view them as three types of objects, ordered by decreasing structure : (bi-ordered) trees can be projected onto texts, and texts can be projected onto words. The other point of view is that words and trees are "inside" a text : they compose the internal structure of a text, in the form of sequential nodes and strictly primitive subtrees of its shape.

We start by taking the first point of view. For given Π and Δ with corresponding ranked alphabet $\Sigma = \Pi \cup \Delta$, the projections involved are the mappings $txt : F_\Sigma \to \mathrm{TXT}_\Pi(\Delta)$, and $word : \mathrm{TXT}_\Pi(\Delta) \to \Delta^+$.

The question addressed here is then : how do these mappings behave with respect to the notions of recognizability and equationality?

Table 1: behaviour with respect to equationality and recognizability

		preserves	reflects
word, *txt*, *yield*	*equat.*	yes	no
	recogn.	no	no
$word^{-1}$, txt^{-1}, $yield^{-1}$	*equat.*	no	yes (if)
	recogn.	yes	yes (if)

For *txt* we obtained some results by applying Theorem 3.4 (see Theorem 5.5). For *word* we can do the same : one may view it as a homomorphism of Σ-algebras, where every operation of Σ of rank $m \geq 2$ is interpreted in Δ^+ as the concatenation of m words.

If $\Sigma_2 \neq \emptyset$, which is the case iff *word* is surjective, then the notions of recognizability and equationality are stable under this extension of the semi-group Δ^+ to a Σ-algebra. Hence in that case Theorem 3.4 can be applied directly. Table 1 presents the results, where we have added the projection from trees to words, $yield : F_\Sigma \to \Delta^+$ (which is defined for an arbitrary ranked alphabet Σ with $\Sigma_0 = \Delta$, see also [11]). Here we say that a mapping *reflects* a property of languages if, given that the image of a language has the property, it follows that the original language has the property. Recall that for tree languages, equationality and recognizability coincide.

In the case that $\Sigma_2 = \emptyset$, then at the places where "(if)" is added the claim is not immediate for the mappings *word* and *yield*; a sufficient condition is that $word(word^{-1}(L)) = L$ and that $yield(yield^{-1}(L)) = L$, respectively.

We now illustrate the no's in the table by giving some examples. First note that the fact that recognizability is not preserved is a consequence of Theorem 3.4(5) and Proposition 3.5, and that by claim (3) of Theorem 3.4 it can be shown that txt^{-1} and $yield^{-1}$ do not preserve equationality.

The first example in Example 7.1 confirms that *txt* and *yield* do not reflect recognizability (nor equationality). The second example shows that also *word* does not reflect recognizability or equationality: a non-context-free text language with a recognizable underlying word language is given. Even if we restrict ourselves a priori to context-free text languages, then still *word* does not reflect recognizability, as is shown in the third example. This example also illustrates that, as opposed to the case of word languages, not all context-free text languages over a one-letter alphabet are recognizable.

Then the only claim left in the above table is that $word^{-1}$ does not preserve equationality, which is shown by the fourth example.

Example 7.1
(1) Let T be the tree language $\{\sigma_f\langle t_\ell^{(n)} t_r^{(n)}\rangle \mid n \geq 1\}$, where $t_\ell^{(n)}$ and $t_r^{(n)}$ are trees inductively defined by $t_\ell^{(1)} = t_r^{(1)} = a$ and for $n > 1$, $t_\ell^{(n)} = \sigma_f\langle t_\ell^{(n-1)} a\rangle$ and $t_r^{(n)} = \sigma_f\langle a t_r^{(n-1)}\rangle$. The tree language T is not recognizable, whereas $txt(T)$ and $yield(T)$ are.
(2) Let π be a primitive bi-order such that $|\pi| = 4$. Let K be the text language that consists of all texts with a shape as sketched in Figure 5.

Then $word(K) = \{a^{6n+4} \mid n \geq 0\}$ is a recognizable word language. Using Proposition 4.10 it can be shown that K is not context-free.

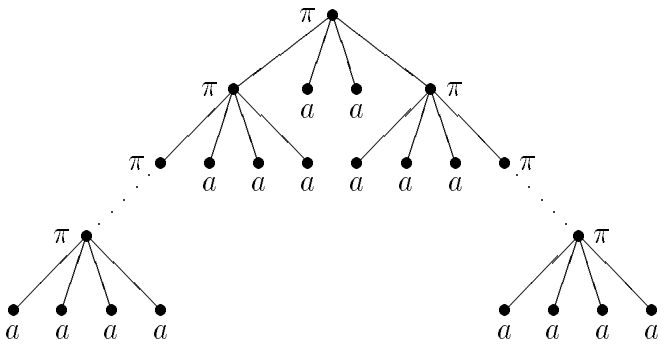


Figure 5: the shapes of a non-context-free text language

(3) Let $G = (N, \Delta, P, \underline{S})$ be a context-free text grammar such that $N = \{S, A, B\}$, $\Delta = \{a\}$, and P consists of the productions $S \to (ASB, (1,2,3))$, $S \to (AB, (1,2))$, $A \to (a^4, (2,4,1,3))$, $B \to (a^5, (2,5,3,1,4))$.

Let $K = \mathrm{TxL}(G)$. Then $word(K) = \{a^{9n} \mid n \geq 1\}$. Clearly, $word(K)$ is a recognizable word language. However, K is not a recognizable text language. We will show this using Lemma 6.1. For $j \geq 1$, we define α_j and β_j as follows. Let $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (2,4,1,3)$, and $(\beta_1, \beta_2, \beta_3, \beta_4, \beta_5) = (2,5,3,1,4)$. If $j = 4k + m$ with $m \in \{1,2,3,4\}$, then $\alpha_j = 4k + \alpha_m$; if $j = 5k + m$ with $m \in \{1,2,3,4,5\}$, then $\beta_j = 5k + \beta_m$. Then we can write K as

$$\{(a^{9n}, (\alpha_1, \ldots, \alpha_{4n}, 4n + \beta_1, \ldots, 4n + \beta_{5n}) \mid n \geq 1\}$$

For $i \geq 1$, let $\tau^{(i)}$ be the text represented by $(a^{5i}, (\beta_1, \ldots, \beta_{5i}))$. Then for all $i, j \geq 1$ with $i \neq j$, there exists a text τ such that $|\tau| = 4i + 1$, $subst(\tau, 4i+1, \tau^{(i)}) \in K$ and $subst(\tau, 4i+1, \tau^{(j)}) \notin K$. Hence for all i, j with $i \neq j$, $\tau^{(i)} \not\cong_K \tau^{(j)}$, which implies that $\not\cong_K$ is not finite. By Lemma 6.1, K is not recognizable.
(4) Let L be the context-free word language $\{a^{3n+2}b^{3n+2} \mid n \geq 0\}$. Let Σ be a ranked alphabet containing the bi-order π from (2). Then the texts with a shape as in Figure 5 but with underlying word $a^{3n+2}b^{3n+2}$ are in $word^{-1}(L)$. It follows that $word^{-1}(L)$ is not a context-free text language, otherwise we could infer from Proposition 4.10 that $word^{-1}(L)$ contains a text τ with $word(\tau) = a^i b^j$ with $i \neq j$. □

In the case of *word*, claim (5) of Theorem 3.4 says that each context-free word language is the projection of a context-free text language. This was noticed in [9], where moreover it was shown that each context-free word language is the projection of a shapely language.

We now take the second point of view, i.e., seeing words and trees as blocks comprising a text. Intuitively, since recognizability and equationality coincide for tree languages, recognizability of context-free text languages depends only on the word parts. To make this observation explicit, we in some way "extract" word languages from a context-free text grammar. Then a text language is context-free, recognizable, shapely iff these "extracted word languages" are context-free, recognizable, finite (Theorem 7.2). This characterization is in particular helpful in the next section, where we consider closure properties.

Formally, we proceed as follows. Let $G = (N, \Delta, P, \tau_0)$ be a context-free text grammar in PNF. Define $V_{p,G}, V_{f,G}, V_{b,G} \subseteq N$ as follows:

$V_{p,G} = \{A \in N \mid \text{there exists } A \to \tau \in P \text{ with } \tau \text{ strictly primitive}\}$

$V_{f,G} = \{A \in N \mid \text{there exists } A \to (w, \sigma_f) \in P\}$

$V_{b,G} = \{A \in N \mid \text{there exists } A \to (w, \sigma_b) \in P\}$

Let i_f and i_b denote the two embeddings from Δ^+ to TXT(Δ) defined by, for $w \in \Delta^+$ with $|w| = n$, $i_f(w) = (w, (1, 2, \ldots, n))$, and $i_b(w) = (w, (n, \ldots, 2, 1))$.

For $x \in \{f, b\}$, define $\Delta_{x,G} = V_{p,G} \cup V_{y,G} \cup \Delta$, where $y \in \{f, b\}$ is such that $y \neq x$, and for $A \in N$, let

$$L_{x,G}(A) = \{w \in \Delta^+_{x,G} \mid \underline{A} \Rightarrow^+ i_x(w)\}$$

Note that if $A \notin V_{x,G}$, then $L_{x,G}(A)$ is empty or a set of singletons. Finally, let

$$\mathcal{L}_G = \{L_{x,G}(A) \mid A \in N, x \in \{f, b\}\}$$

Theorem 7.2 *Let K be a text language.*
(1) K is context-free iff there exists a context-free text grammar G in PNF generating K such that each $L \in \mathcal{L}_G$ is a context-free word language.
(2) K is recognizable iff there exists a context-free text grammar G in PNF generating K such that each $L \in \mathcal{L}_G$ is a recognizable word language.
(3) K is shapely iff there exists a context-free text grammar G in PNF generating K such that each $L \in \mathcal{L}_G$ is a finite word language.

Proof. We will use the notations given above, where we omit the subscript G if the context-free text grammar G is clear from the context. Note that for the proof it suffices to consider the languages $L \in \mathcal{L}_G$ of the form $L_x(A)$ with $A \in V_x$.

(1) The if-part is, of course, trivial. Let K be a context-free text language, and let $G = (N, \Delta, P, \tau_0)$ be a context-free text grammar in PNF for K. Consider $L \in \mathcal{L}_G$. Suppose that $L = L_f(A)$ for some $A \in V_f$. We show that L is a context-free word language by giving a context-free word grammar for L. The set of nonterminals is $N_f = \{A^f \mid A \in V_f\}$ and the set of productions is

$$P_{f,A} = \{A^f \to w \mid A \to i_f(w') \in P, w \in \varphi(w')\}$$

where φ is the substitution such that for $B \in N \cup \Delta$,

$$\varphi(B) = \begin{cases} \{B^f\} & B \in V_f - (V_p \cup V_b) \\ \{B^f, B\} & \text{if } B \in V_f \cap (V_p \cup V_b) \\ \{B\} & \text{otherwise} \end{cases}$$

Then for the context-free word grammar $G' = (N_f, \Delta_f, P_{f,A}, A^f)$, $L(G') = L_f(A) = L$.

Similarly, if $L = L_b(A)$ for some $A \in V_b$, then L is a context-free word language.

(2) Let K be a recognizable text language, and let G be a right-linear context-free text grammar for K. Then the context-free grammar G' constructed for $L \in \mathcal{L}_G$ as in (1) is a right-linear word grammar. Hence $L = L(G')$ is a recognizable word language.

Suppose that G is a context-free text grammar for K such that each $L \in \mathcal{L}_G$ is recognizable. For each $L \in \mathcal{L}_G$, let G_L be a right-linear context-free word grammar in

Chomsky normal form with production-set P_L. These grammars G_L can be chosen in such a way that the following conditions are satisfied :
- the axiom of G_L is A if $L = L_x(A)$, $x \in \{b, f\}$,
- A does not occur in any right-hand side of P_L, and
- the remaining nonterminals of G_L are disjoint from those of $G_{L'}$ for all $L' \in \mathcal{L}_G$, and disjoint from N.

We remove all sequential productions from G, and add the productions
$\{X \to i_f(w) \mid X \to w \in P_L, L = L_f(A) \text{ for some } A\}$ and
$\{X \to i_b(w) \mid X \to w \in P_L, L = L_b(A) \text{ for some } A\}$.
The thus obtained context-free text grammar is right-linear and equivalent to G. Hence, by Theorem 6.9, K is recognizable.

(3) Let K be a text language, and let G be a context-free text grammar in PNF for K. Let $\tau \in K$, and let t be a derivation tree of τ in G. By Proposition 4.5, $di(t)$ is a refinement of the shape of τ. It follows that for each sequential node of the shape of τ, say with n children, there is a corresponding derivation $\underline{A} \Rightarrow^+ i_x(w)$ in G such that $|w| = n$ and $w \in L_x(A)$.

Conversely, if w is a word of length n in some $L_x(A) \in \mathcal{L}_G$, then there is a derivation tree t in G of a text $\tau \in K$ with the following property : there is a subtree of t which is a derivation tree of $i_x(w)$ from A and moreover this subtree corresponds with a sequential node with n children in the shape of τ.

Hence the outdegrees of the sequential nodes in the shapes of K are precisely the lengths of the words occurring in the languages $L \in \mathcal{L}_G$. It follows that K is limited iff every $L \in \mathcal{L}_G$ is finite. By Proposition 4.9, this proves (3). □

In (1) and (3) of the proof it is in fact shown that K is context-free (shapely) iff for each context-free text grammar G in PNF generating K each $L \in \mathcal{L}_G$ is a context-free (finite) word language. Concerning (2), it is shown that K is recognizable iff for each *right-linear* grammar G generating K each $L \in \mathcal{L}_G$ is a recognizable word language; here we can not replace "right-linear" by "in PNF" as Example 7.3 will show.

Example 7.3 Let $G = (N, \Delta, P, (S, \sigma_1))$ be a context-free text grammar such that P consists of the productions $S \to (ASB, (1,2,3))$, $S \to (AB, (1,2))$, $A \to (a^4, (2,4,1,3))$, and $B \to (a^4, (2,4,1,3))$.

Then $\mathrm{TxL}(G)$ is recognizable. However, $L_f(S) = \{A^n B^n \mid n \geq 1\}$ is a non-recognizable word language. □

8 Closure properties

Most operations on text languages given in this section are defined for arbitrary text languages, i.e., with possibly infinite set of operations $op(K)$, except for the "algebraic closure", which must be defined w.r.t. a fixed text algebra $\mathcal{T}_\Sigma$.

First we introduce the operations which will provide an operational characterization of the context-free text languages (Theorem 8.4). These operations are the natural extensions of substitution and substitution closure on word languages.

Definition 8.1 A mapping $j : \mathrm{TXT}(\Delta) \to 2^{\mathrm{TXT}(\Delta)}$ is an *alphabetic (text) substitution (on Δ)* if there is a mapping $j_0 : \Delta \to 2^{\mathrm{TXT}(\Delta)}$ such that for $\tau = (a_1 \cdots a_m, \sigma)$,

$$j(\tau) = \{[\sigma \leftarrow (\tau_1, \ldots, \tau_m)] \mid \tau_i \in j_0(a_i) \text{ for } i = 1, \ldots, m\}$$

j is a *unary* alphabetic substitution *at a* if $j_0(x) = \{x\}$ for every $x \in \Delta$ with $x \neq a$.

We will use j to denote both the mapping j and the mapping j_0. The adjective *alphabetic* is used to distinguish this notion of substitution from the singular and simultaneous substitutions on texts as defined in Section 4.

Definition 8.2 Let $a \in \Delta$ and let $K \subseteq \mathrm{TXT}(\Delta)$. The *alphabetic substitution closure of* K *at* a is defined to be $\bigcup_{n=0}^{\infty} j^n(\{a\})$, where j is the unary alphabetic substitution at a such that $j(a) = K \cup \{a\}$.

Alphabetic text substitution and alphabetic text substitution closure are the counterparts of the regular operations on tree languages : tree concatenation and tree concatenation closure (called "forest products" in [11]). We now give the counterparts of the regular operations on word languages.

The operations given by the bi-orders of PRIM generalize concatenation of word languages (and correspond with so-called "top-concatenation" of tree languages). Let $\sigma \in \mathrm{PRIM}$ with rank $m \geq 2$. For text languages $K_1, \ldots, K_m \subseteq \mathrm{TXT}(\Delta)$, $[\sigma \leftarrow (K_1, \ldots, K_m)]$ is the text language $\{[\sigma \leftarrow (\tau_1, \ldots, \tau_m)] \mid \tau_i \in K_i \text{ for } i = 1, \ldots, m\}$.

By the *algebraic closure* (*w.r.t.* $\Sigma = \Pi \cup \Delta$) *of* a text language $K \subseteq \mathrm{TXT}_\Pi(\Delta)$ we mean the language $\{[\sigma \leftarrow (\tau_1, \ldots, \tau_m)] \mid (w, \sigma) \in \mathrm{TXT}_\Pi(\Delta)$ for some $w \in \Delta^+$ with $|w| = m,\ \tau_1, \ldots, \tau_m \in K\}$, i.e., the sub-algebra of $\mathcal{T}_\Sigma$ generated by K. Algebraic closure generalizes Kleene closure of word languages.

Theorem 8.3
(1) CFT *is closed under union, the operations of* PRIM*, algebraic closure, alphabetic substitution, alphabetic substitution closure, and intersection with recognizable text languages.*
(2) CFT *is not closed under intersection and complement.*

Proof. (1) By standard constructions as in [11, Ch. II-4], and [14, Ch. I-3]. See also [4], where in particular it is shown that the intersection of an equational and a recognizable set is again equational (w.r.t. to an arbitrary Σ-algebra).

As an example we will give the proof for the operations of PRIM and for the alphabetic substitution operation.

Let $\sigma \in \mathrm{PRIM}$ with $|\sigma| = m$, and let $K_1, \ldots, K_m \in \mathrm{CFT}$ be such that $K_i = \mathrm{TxL}(G_i)$ with $G_i = (N_i, \Delta_i, P_i, \underline{S_i})$ for $i = 1, \ldots, m$. Let $K = [\sigma \leftarrow (K_1, \ldots, K_m)]$. K is generated by the context-free text grammar $G = (N, \Delta, P, \underline{S})$, where $S \notin \bigcup_{i=1}^m N_i$, $N = (\bigcup_{i=1}^m N_i) \cup \{S\}$, $\Delta = \bigcup_{i=1}^m \Delta_i$, and $P = (\bigcup_{i=1}^m P_i) \cup \{S \rightarrow (S_1 \ldots S_m, \sigma)\}$. Hence $K = \mathrm{TxL}(G) \in \mathrm{CFT}$.

Let $K \in \mathrm{CFT}$ be a text language over Δ, and let j be an alphabetic substitution on Δ. For $a \in \Delta$, let $G_a = (N_a, \Delta, P_a, \underline{S_a})$ be a context-free text grammar in PNF for $j(a)$ such that if $a \neq b$, then $N_a \cap N_b = \emptyset$. Let $G = (N, \Delta, P, \underline{S})$ be a context-free grammar in CPNF for K such that N is disjoint from each N_a. Now $G' = (\bigcup_{a \in \Delta} N_a \cup N, \Delta, P', \underline{S})$ is a context-free text grammar generating $j(K)$, where $P' = (\bigcup_{a \in \Delta} P_a) \cup \{A \rightarrow \tau \in P \mid |\tau| \geq 2\} \cup \{A \rightarrow \tau \mid A \rightarrow \underline{a} \in P, S_a \rightarrow \tau \in P_a\}$.

Hence $\mathrm{TxL}(G') = j(K)$ is a context-free text language, i.e. $j(K) \in \mathrm{CFT}$.

(2) Let L_1, L_2 be context-free word languages such that $L_1 \cap L_2$ is not context-free. Consider the corresponding forward sequential text languages, i.e., $K_1 = i_f(L_1)$ and $K_2 = i_f(L_2)$. Then K_1 and K_2 are context-free text languages, and $word(K_1 \cap K_2) = word(K_1) \cap word(K_2)$ (note that this is not generally true for arbitrary context-free text languages K_1,K_2). Then $K_1 \cap K_2$ is a text language which is *not* context-free, otherwise

$word(K_1 \cap K_2) = L_1 \cap L_2$ would be a context-free word language. For the complement a similar argument applies. □

The following theorem gives an operational characterization of context-free text languages. It is a consequence of Theorem 8.3 and the fact that each context-free text language can be obtained from finite text languages by union, alphabetic substitution, and alphabetic substitution closure, which can be shown by Theorem 5.5(1) and the analogous result for tree languages (Theorem 5.8 in [11], where tree languages obtained from finite tree languages using the analogous operations are called "regular"), or by directly performing a similar construction as in the proof given there in terms of texts.

Theorem 8.4 CFT *is the smallest family of text languages containing the finite text languages that is closed under union, alphabetic substitution, and alphabetic substitution closure.* □

It is well-known (and easy to prove) that recognizable subsets (w.r.t. to any Σ-algebra) are closed under union, intersection, and complement.

In [4] closure properties of recognizable subsets of algebras over a theory are investigated, depending on the form of the equations in the theory. From this we obtain that for each so-called "relabeling" $r : \Delta \to \Gamma$, if $K \subseteq \mathrm{TXT}(\Delta)$ is recognizable, then $r(K) \subseteq \mathrm{TXT}(\Gamma)$ is recognizable. Note that a relabeling is a special case of alphabetic substitution. We will show that in our specific case of texts, RECT is closed under the operations of PRIM, algebraic closure, and alphabetic substitution. This is a consequence of the fact that recognizable *word* languages are closed under (word) concatenation, Kleene plus, and (word) substitution, respectively.

Theorem 8.5
(1) RECT *is closed under union, intersection, complement, the operations of* PRIM*, algebraic closure, and alphabetic substitution.*
(2) RECT *is not closed under alphabetic substitution closure.*

Proof. (1) Let $\sigma \in \mathrm{PRIM}$ with $|\sigma| = m \geq 2$. Let $K_1, \ldots, K_m$ be recognizable text languages, and let $K = [\sigma \leftarrow (K_1, \ldots, K_m)]$. By Theorem 7.2, for each $i \in \{1, \ldots, m\}$, there is a context-free text grammar $G_i = (N_i, \Delta_i, P_i, \underline{S_i})$ in PNF generating K_i such that each $L \in \mathcal{L}_{G_i}$ is a recognizable word language.

Let G be the text grammar from the proof of Theorem 8.3 that generates K. We will use the notations from Section 7 used in Theorem 7.2. Let $A \in V_{f,G}$. If $A \in V_{f,G_i}$ for some $i \in \{1, \ldots, m\}$, then $L_{f,G}(A) = L_{f,G_i}(A)$, which is a recognizable word language by Theorem 7.2. The case that $A \notin V_{f,G_i}$ for each $i \in \{1, \ldots, m\}$ occurs iff $\sigma = \sigma_f$ and $A = S$; then $L_{f,G}(S) = L_{f,G_1}(S_1) \cdot L_{f,G_2}(S_2)$. Since $L_{f,G_1}(S_1)$ and $L_{f,G_2}(S_2)$ are recognizable word languages by Theorem 7.2, and recognizable word languages are closed under concatenation, it follows that $L_{f,G}(S)$ is recognizable.

Similarly, we show that the word languages, $L_{b,G}(A)$, $A \in V_{b,G}$, are recognizable.

Consequently, each $L \in \mathcal{L}_G$ is recognizable, and it follows by Theorem 7.2 that $\mathrm{TxL}(G) = K$ is a recognizable text language.

This proves that RECT is closed under the operations of PRIM.

Let K be a recognizable text language, generated by the grammar $G = (N, \Delta, P, \underline{S})$. Let $\Sigma = \Pi \cup \Delta$ be a ranked alphabet such that $K \subseteq \mathrm{TXT}_\Pi(\Delta)$. Let $G' = (N, \Delta, P', \underline{S})$, where $P' = P \cup \{S \to (S^m, \sigma) \mid \sigma \in \Pi, |\sigma| = m\}$. Then $\mathrm{TxL}(G')$ is the algebraic closure

of K w.r.t. Σ. For $x \in \{f, b\}$, $L_{x,G'}(A) = L_{x,G}(A)$ for all $A \neq S$, and $L_{x,G'}(S) = L_{x,G}(S)$ if $\sigma_x \notin \Sigma$, $L_{x,G'}(S) = (L_{x,G}(S))^+$ otherwise. Hence $\mathcal{L}_{G'}$ consists of recognizable word languages, which implies that $\mathrm{TxL}(G')$ is a recognizable text language. Hence RECT is closed under algebraic closure.

Now let j be an alphabetic substitution. Let G' be the text grammar in PNF from the proof of Theorem 8.3 that generates $j(K)$. Let $x \in \{f, b\}$, and let $A \in V_{x,G'}$. If $A \in N_a$ for some $a \in \Delta$, then $A \in V_{x,G_a}$, and $L_{x,G'}(A) = L_{x,G_a}(A)$ is a recognizable word language. If $A \in N$, then $L_{x,G'}(A) = j_x(L_{x,G}(A))$, where j_x is the word substitution on $N \cup \Delta$ defined by $j_x(A) = A$ for each $A \in N$ and $j_x(a) = L_{x,G_a}(S_a)$ for each $a \in \Delta$. Since, by Theorem 7.2, the word languages of the form $L_{x,G_a}(S_a)$ are recognizable and recognizable word languages are closed under substitution, it follows that $L_{x,G'}(A)$ is recognizable.

Hence each $L \in \mathcal{L}_{G'}$ is recognizable, and $K = \mathrm{TxL}(G')$ is a recognizable text language. Consequently, RECT is closed under alphabetic substitution.

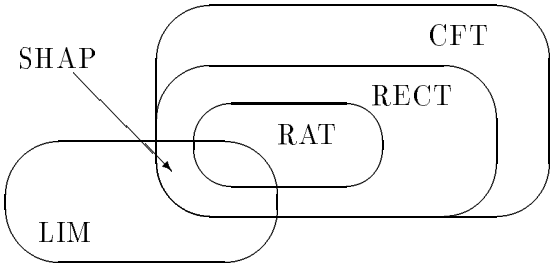
(2) This follows immediately from the fact that RECT $\subset$ CFT combined with (1) and Theorem 8.4. □

One could also prove (2) using the fact that recognizable word languages are not closed under substitution closure.

For recognizable text language we do not have a characterization as in Theorem 8.4. The operations derived from regular word operations do not characterize the recognizable text languages, but the *rational* text languages. In [15], rational subsets of arbitrary Σ-algebras are defined as those subsets built from finite languages by union, the operations of Σ and algebraic closure w.r.t. Σ. A general property of rational sets in an arbitrary algebra (cf. Theorem 3.4) is that the homomorphic image of a rational set is rational. It may happen that the class of recognizable sets is strictly contained in the class of rational sets (e.g., in arbitrary monoids), or that the class of rational sets is strictly contained in the class of recognizable sets (as shown for tree languages in [15]). By Kleene's Theorem, the rational word languages are precisely the recognizable word languages.

For text languages we have by Theorem 8.5 that RAT $\subseteq$ RECT, where RAT is the class of rational text languages. Considering the homomorphism *word* as in Section 7, we obtain, by Kleene's Theorem, that the underlying word languages of rational text languages are recognizable (cf. the case of recognizable text languages, where the underlying word languages are context-free). This shows that RAT $\subset$ RECT.

Summarizing, we can extend Figure 4 yielding the inclusion diagram in Figure 6.



æ

Figure 6: families of text languages

Acknowledgements

The authors are indebted to A. Ehrenfeucht and G. Rozenberg for encouraging the research that led to this paper, and to an anonymous referee for making helpful comments.
æ

References

[1] P.M. Cohn, *Universal Algebra*, Harper & Row, New York, 1965.

[2] B. Courcelle, Equivalences and transformations of regular systems; applications to recursive program schemes and grammars, *Theoret. Comput. Sci.* **42** (1986) 1–122.

[3] B. Courcelle, An axiomatic definition of context-free rewriting and its application to NLC graph grammars, *Theoret. Comput. Sci.* **55** (1988) 141–181.

[4] B. Courcelle, On recognizable sets and tree automata, in *Resolution of Equations in Algebraic Structures, Vol 1*, H. Ait-Kaci and M. Nivat, eds., Academic Press, New York, 1989.

[5] A. Ehrenfeucht and G. Rozenberg, Theory of 2-structures, Part I: clans, basic subclasses, and morphisms, *Theoret. Comput. Sci.* **70** (1990) 277–303.

[6] A. Ehrenfeucht and G. Rozenberg, Theory of 2-structures, Part II: representation through labeled tree families, *Theoret. Comput. Sci.* **70** (1990) 305–342.

[7] A. Ehrenfeucht and G. Rozenberg, T-functions, T-structures, and texts, *Theoret. Comput. Sci.* **116** (1993) 227–290.

[8] A. Ehrenfeucht, P. ten Pas, and G. Rozenberg, Combinatorial properties of texts, *RAIRO, Theor. Inf.* **27** (1993) 433–464.

[9] A. Ehrenfeucht, P. ten Pas, and G. Rozenberg, Context-free text grammars, *Acta Informatica* **31** (1994) 161–206.

[10] A. Ehrenfeucht, H.J. Hoogeboom, P. ten Pas, and G. Rozenberg, An introduction to context-free text grammars, in *Developments in Language Theory*, G. Rozenberg and A. Salomaa, eds., World Scientific Publishing, Singapore, 1994, 357–369.

[11] F. Gecseg and M. Steinby, *Tree Automata*, Akademiai Kiado, Budapest, 1984.

[12] J. Mezei and J.B. Wright, Algebraic automata and context-free sets, *Information and Control* **11** (1967) 3–29.

[13] W.C. Rounds, Context free grammars on trees, in *Proceedings ACM STOC '69*, ACM, New York, 1969, 143–148.

[14] A. Salomaa, *Formal Languages*, Academic Press, New York, 1973.

[15] M. Steinby, Some algebraic aspects of recognizability and rationality, in *Proceedings FCT '81*, F. Gecseg, ed., LNCS 117, Springer Verlag, Berlin, 1981, 360–372.