# An algebraic approach to Multiple Context-Free Grammars

Alexander Clark[1] and Ryo Yoshinaka[2]

[1] Department of Philosophy, King's College London
alexander.clark@kcl.ac.uk
[2] Graduate School of Informatics, Kyoto University
ry@i.kyoto-u.ac.jp

**Abstract.** We define an algebraic structure, Paired Complete Idempotent Semirings (PCIS), which are appropriate for defining a denotational semantics for multiple context-free grammars of dimension 2 (2-MCFG). We demonstrate that homomorphisms of this structure will induce well-behaved morphisms of the grammar, and generalize the syntactic concept lattice from context-free grammars to the 2-MCFG case. We show that this lattice is the unique minimal structure that will interpret the grammar faithfully and that therefore 2-MCFGs without mergeable nonterminals will have nonterminals that correspond to elements of this structure.

## 1 Introduction

Denotational semantics[3] have been provided before for MCFGs and indeed for richer formalisms [6, 7] but they have never specified precisely what the operations or algebraic structures that provide are. When defining denotational semantics for CFGs [5] this is well known: the appropriate structure is a complete idempotent semiring (CIS) [4]. Recently it has been shown [2] that homomorphisms of this structure induce nice grammar morphisms and that there is a unique smallest structure that interprets the CFG faithfully. As a result, CFGs without mergeable nonterminals will have nonterminals that correspond to elements of this structure. This structure, the Syntactic Concept Lattice, is also very important from the perspective of grammatical inference, as it gives rise to a number of algorithms for learning grammars using *distributional* techniques [1, 12]. It consists of the collection of all sets of strings that are distributionally definable with respect to the language in question.

Given the central importance of mildly context-sensitive grammars in mathematical linguistics, it is clearly desirable to extend these techniques beyond the class of CFGS; here we take the next step to the class of 2-MCFGs. For a 2-MCFG, unsurprisingly, we need a richer structure than a CIS; which we define in this paper, and which we call a Paired Complete Idempotent Semiring (PCIS). We add a tupling operation and some nontrivial additional axioms. We then define the

[3] Here we refer to the semantics of the meta-grammar rather than the meanings of natural language sentences [8].

natural extension of the Syntactic Concept Lattice, which we call the Syntactic Concept Lattice of order 2, which consists of all distributionally definable sets of strings and sets of pairs of strings, where the notion of distribution is extended appropriately to account for the richer set of derivation contexts in a 2-MCFG. As a result we can show that every grammar without mergeable nonterminals for a language defined by a 2-MCFG will have nonterminals that correspond to the elements of this concept lattice.

A brief roadmap for the paper: we start (Section 2) by defining MCFGs and morphisms of these grammars in a fairly standard way. These grammars use a certain (infinite) family of linear regular functions. We then define in Section 3 the algebraic structures, Paired Complete Idempotent Semirings (PCIS) that we use to interpret these grammars, and show (Section 4) how can define the relevant linear regular functions in this structure. At this point we also define the standard Ginsburg and Rice-style fixed point semantics for 2-MCFGs. The generalization of the syntactic concept lattice is then defined in Section 5. At this point we can unify these two rather independent strands by showing that PCIS-homomorphisms induce well-behaved MCFG-morphisms, and that in particular the homomorphism into the syntactic concept lattice gives a grammar morphism that does not increase the language (Section 6). Moreover we show that it is the unique up to isomorphism simplest structure with this property.

## 2 Definitions

We assume a fixed nonempty finite set as an alphabet, $\Sigma$, and we write $\Sigma^*$ for the set of finite strings over $\Sigma$, and $\lambda$ for the empty string. We write $\mathcal{P}(X)$ for the power set of a set $X$. A language is a subset of $\Sigma^*$. Given two languages $A, B$ we define $A \cdot B$ to be the standard concatenation of languages, and $A \times B$ to be the Cartesian product.

We recall the definition of a complete idempotent semiring (CIS), which is a tuple

$$\langle M, \circ, \vee, \epsilon, \bot \rangle$$

where $M$ is a set, $\epsilon, \bot \in M$, $\circ$ is a binary operation such that $\langle M, \circ, \epsilon \rangle$ is a monoid, $\vee$ is an commutative idempotent operation with $\bot$ as identity, together with the infinitary variant of $\vee$ and so where $\langle M, \vee, \bot \rangle$ is a complete semilattice with zero. Moreover we require that $x \circ \bot = \bot = \bot \circ x$ for all $x \in M$ and that $\circ$ distributes left and right over $\vee$ so that for any $Y \subseteq M$, $x \circ \bigvee Y = \bigvee \{x \circ y \mid y \in Y\}$ and $\bigvee Y \circ x = \bigvee \{y \circ x \mid y \in Y\}$.

Given the $\vee$ operation we can naturally define a partial order $\leq$ by $x \leq y$ iff $y = x \vee y$.

The free CIS over a finite set $\Sigma$ is defined to be

$$\langle \mathcal{P}(\Sigma^*), \cdot, \cup, \{\lambda\}, \emptyset \rangle$$

which we can easily verify to be a CIS. These provide the standard denotational semantics for CFGS [5, 4].

For reasons of space, we assume that the reader is familiar with the MCFG formalism [10]. We denote an MCFG by $G = \langle \Sigma, V, S, P \rangle$, where $\Sigma$ is the set of terminal symbols, $V$ is the set of nonterminal symbols, $S$ is the start symbol and $P$ is the set of production rules. Each nonterminal in $V$ is assigned a positive integer called *dimension* and we let $V_d$ be the set of nonterminals of dimension $d$. We write $\dim(N) = d$ if $N \in V_d$. Each nonterminal of dimension $k$ derives $k$-tuples of strings. The start symbol $S$ has dimension 1, so it generates usual strings. There are different ways to describe rules of $P$.

For example, the rules of a grammar $G = \langle \Sigma, V, S, P \rangle$ can be described as

$$S(z_{1,1}z_{2,1}z_{1,2}z_{2,2}) \coloneq N(z_{1,1}, z_{1,2}), M(z_{2,1}, z_{2,2})\,; \quad N(a, c)\,; \quad M(b, d)\,;$$
$$N(z_{1,1}a, z_{1,2}c) \coloneq N(z_{1,1}, z_{1,2})\,; \quad M(z_{1,1}b, z_{1,2}d) \coloneq M(z_{1,1}, z_{1,2})\,,$$

where each $z_{i,j}$ is a variable representing the $j$th element of the $i$th argument. In the above example, $N(a, c) \in P$ means that $N$ generates a pair $(a, c)$. The rule $S(z_{1,1}z_{2,1}z_{1,2}z_{2,2}) \coloneq N(z_{1,1}, z_{1,2}), M(z_{2,1}, z_{2,2})$ means that if $N$ generates a pair $(z_{1,1}, z_{1,2})$ and $M$ generates $(z_{2,1}, z_{2,2})$ then $S$ generates the string of the form $z_{1,1}z_{2,1}z_{1,2}z_{2,2}$ by concatenating these strings. This grammar generates the language $\{\, a^m b^n c^m d^n \mid m, n \geq 1 \,\}$. In general, a rule has the following form:

$$N_0(\alpha_1, \dots, \alpha_{d_0}) \coloneq N_1(z_{1,1}, \dots, z_{1,d_1}), \dots, N_m(z_{m,1}, \dots, z_{m,d_m})$$

where $\dim(N_i) = d_i$ for $i = 0, \dots, m$ and $\alpha_k \in (\Sigma \cup \{\, z_{i,j} \mid 1 \leq i \leq m,\ 1 \leq j \leq d_i \,\})^*$ for $k = 1, \dots, d_0$ such that no variable $z_{i,j}$ occurs in $\alpha_1 \dots \alpha_{d_0}$ twice or more.

Alternatively, the above rules can be written using *linear functions*:

$$S \to f(N, M),\ N \to \langle a, c \rangle,\ M \to \langle b, d \rangle,\ N \to g(N),\ M \to h(M),$$

where $f, g, h$ are linear functions satisfying that for any $u_{i,j} \in \Sigma^*$

$$f(\langle u_{1,1}, u_{1,2} \rangle, \langle u_{2,1}, u_{2,2} \rangle) = u_{1,1}u_{2,1}u_{1,2}u_{2,2}\,,$$
$$g(\langle u_{1,1}, u_{1,2} \rangle) = \langle u_{1,1}a, u_{1,2}c \rangle\,,\ h(\langle u_{1,1}, u_{1,2} \rangle) = \langle u_{1,1}b, u_{1,2}d \rangle\,.$$

We often identify a linear function and its specification as a tuple of strings of terminals and variables. In the above examples, $g$ is identified with $\langle z_{1,1}a, z_{1,2}c \rangle$.

We write the set of $\dim(N)$-tuples of strings generated by a nonterminal $N$ of $G$ by $\mathcal{L}(G, N) \subseteq (\Sigma^*)^{\dim(N)}$, which is recursively defined as follow: if a rule $N_0 \to f(N_1, \dots, N_m)$ is in $P$ and $\boldsymbol{u}_i \in \mathcal{L}(G, N_i)$ for $i = 1, \dots, m$, then $f(\boldsymbol{u}_1, \dots, \boldsymbol{u}_m) \in \mathcal{L}(G, N_0)$. The language of $G$ is $\mathcal{L}(G) = \mathcal{L}(G, S)$.

A linear function uses each element of arguments at most once. We say that a linear function is *regular* or *non-deleting* if every argument appears as a part of the output. Moreover a linear regular function is *non-permuting* if $z_{i,j+1}$ never occurs left to $z_{i,j}$ in its specification.

In this paper, we assume that all linear functions are non-deleting and non-permuting and moreover those functions introduce terminal symbols only when the arity is 0, so their specifications are either variable-free or terminal-free. It is known that those restrictions on linear functions do not affect the generative power of the MCFG formalism. In addition, we concern ourselves here with MCFGs whose nonterminals have dimension at most 2, i.e, $V = V_1 \cup V_2$. We define this class of grammars as 2-MCFG.

## 3 Paired Complete Idempotent Semirings

We use a multi-sorted algebra which has two types of elements: roughly sets of strings (sort 1) and sets of pairs of strings (sort 2).

**Definition 1.** *A Paired Complete Idempotent Semiring (* PCIS*) is a tuple*

$$\langle A_1, A_2, \circ, \otimes, \vee^1, \vee^2, \epsilon, \bot \rangle$$

*where $A_1, A_2$ are two disjoint*[4] *sets, and $\epsilon, \bot \in A_1$. We call these the sets of* elements of sort 1 and sort 2 *respectively. We also have operations defined on elements of various sorts, $\vee^1, \vee^2$ which are commutative and idempotent, together with their infinitary variants, $\bigvee^1, \bigvee^2$, a binary concatenation operation $\circ$ on $A_1$, and a pairing operation that constructs elements of sort 2 from two elements of sort 1, $\otimes$. We will often drop the sortal superscripts on $\vee$ as they are clear from the context and overload the symbol $\vee$ or $\bigvee$.*

*These satisfy the following axioms:*

- $\langle A_1, \circ, \vee^1, \epsilon, \bot \rangle$ *is a* CIS*;*
- $\langle A_2, \vee^2, \bot \otimes \bot \rangle$ *is a complete join semi-lattice with bottom element;*
- $\vee^1$ *and* $\vee^2$ *distribute over* $\otimes$ *that is to say we have the following identities:*
  - $(x \vee^1 y) \otimes z = (x \otimes z) \vee^2 (y \otimes z)$ *and* $(\bigvee_i x_i) \otimes y = \bigvee_i (x_i \otimes y)$ ,
  - $x \otimes (y \vee^1 z) = (x \otimes y) \vee^2 (x \otimes z)$ *and* $x \otimes (\bigvee_i y_i) = \bigvee_i (x \otimes y_i)$ .

*We define partial orders on the two sorts, $x \leq_1 y$ iff $y = x \vee^1 y$, $x \leq_2 y$ iff $y = x \vee^2 y$. Note that $\bot_1$ is a minimal element with respect to $\leq_1$ and $\bot_2 = \bot_1 \otimes \bot_1$ with respect to $\leq_2$.*

**Axiom E***: For all $x \in A_2$,*

$$x = \bigvee^2 \{u \otimes v \mid u, v \in A_1, u \otimes v \leq_2 x\} .$$

**Axiom V***: For any set $J \subseteq A_1 \times A_1$, and any $p, q, a, b, c, d \in A_1$ if*

$$p \otimes q \leq_2 \bigvee \{u \otimes v \mid (u, v) \in J\}$$

*then*

$$p \circ q \leq_1 \bigvee \{u \circ v \mid (u, v) \in J\}$$

*and*

$$(a \circ p \circ b) \otimes (c \circ q \circ d) \leq_2 \bigvee \{(a \circ u \circ b) \otimes (c \circ v \circ d) \mid (u, v) \in J\} .$$

[4] We allow a slight violation of this criterion later as the empty set may be a member of both sets. This can be resolved with some technical modifications or notational variant.

Axioms E and V require some explanation. Axiom E excludes the case where there may be elements of sort 2 that are not formed from elements of sort 1. Axiom V can best be thought of as a generalization of the other distributivity axioms. From the CIS axioms we have that $\circ$ distributes over $\vee^1$. From Axiom V we can show that the general family of intercalation operations that we use in MCFG derivations distribute over $\vee_1$ and $\vee_2$ using the two clauses of the axiom respectively.

We use repeatedly the convention that $d$ is a variable ranging over 1 and 2, which allows us to state conditions that both sorts satisfy. If $A$ is a PCIS, then we write $A_d$ for the set of elements of sort $d$, $\bot_A$ or $\bot_1$ for the bottom element of $A_1$ and $\epsilon_A$ for the $\circ$-identity.

The most basic type of PCIS which will help to understand the definition is the free structure over a finite set which will be very familiar.

**Definition 2.** *Given a finite set $\Sigma$ we define the free structure $\mathfrak{F}(\Sigma)$ to be the structure:*

$$\langle \mathcal{P}(\Sigma^*), \mathcal{P}(\Sigma^* \times \Sigma^*), \cdot, \times, \cup, \cup, \{\lambda\}, \emptyset \rangle$$

The elements of sort 1 are just sets of strings, the elements of sort 2 are sets of pairs of strings, the pairing operation is the cartesian product and so on.

We define homomorphisms standardly.

**Definition 3.** *Given two PCISs $A$ and $B$, a PCIS-homomorphism $h$ is a pair of functions $h_1 : A_1 \to B_1$ and $h_2 : A_2 \to B_2$ such that:*

- $h_1(\bot_A) = \bot_B$
- $h_1(\epsilon_A) = \epsilon_B$
- *For all* $u, v \in A_1$, $h_1(u \circ v) = h_1(u) \circ h_1(v)$
- *For all* $u, v \in A_1$, $h_2(u \otimes v) = h_1(u) \otimes h_1(v)$
- *For all* $X \subseteq A_d$, $h_d(\bigvee^d X) = \bigvee^d \{h_d(x) \mid x \in X\}$

**Definition 4.** *Suppose we have a PCIS-homomorphism $h : A \to B$; we define $h^* : B \to A$ as a pair of functions $h^*_d : B_d \to A_d$ with $d = 1, 2$, where for all $y \in B_d$ we define*

$$h^*_d(y) = \bigvee^d \{x \in A_d \mid h_d(x) \leq y\} .$$

This function $h^*$ is called the residual of the map $h$, which is a weak sort of inverse.

## 4 Algebraic Semantics for MCFGs

The PCIS only has the single concatenation operation $\circ$, but we can define linear regular functions in a natural way. We will illustrate this with a simple example. Consider the MCFG production

$$N(z_{1,1}z_{2,1}, z_{2,2}z_{1,2}) :- P(z_{1,1}, z_{1,2}), Q(z_{2,1}, z_{2,2})$$

which uses the linear regular function

$$f((z_{1,1}, z_{1,2}), (z_{2,1}, z_{2,2})) = (z_{1,1}z_{2,1}, z_{2,2}z_{1,2})\,.$$

We can extend these functions naturally from strings into a PCIS.

**Definition 5.** *For all $x, y \in A_2$ we define $x \ominus y$ to be*

$$x \ominus y = \bigvee \{(z_{1,1} \circ z_{2,1}) \otimes (z_{2,2} \circ z_{1,2}) \mid (z_{1,1} \otimes z_{1,2}) \leq x, (z_{2,1} \otimes z_{2,2}) \leq y\}$$

*where the $z$ variables range over elements of $A_1$. This defines a binary operation on $A_2$.*

Here we implicitly rely on Axiom E; we apply the linear regular function to all of the sort 1 components of $x$ and $y$, and then combine the results with $\bigvee$. Crucially, these operations, given the axioms of the PCIS, are "homomorphic".

**Lemma 1.** *If $h$ is a PCIS-homomorphism then $h(x \ominus y) = h(x) \ominus h(y)$.*

*Proof.* (Sketch) It is immediate from the definitions that $h(x \ominus y) \leq h(x) \ominus h(y)$. Now $h(x) \ominus h(y) = \bigvee Z$ where $Z = \{(z_{1,1} \circ z_{2,1}) \otimes (z_{2,2} \circ z_{1,2}) \mid (z_{1,1} \otimes z_{1,2}) \leq h(x), (z_{2,1} \otimes z_{2,2}) \leq h(y)\}$. Using Axiom V we can show that every element of $Z$ is bounded by $h(x \ominus y)$, which gives the result. □

In a similar way we can define for any linear regular function $f$, a corresponding operation on a PCIS. These operations coincide in the trivial cases with primitive operations: for example, if $f$ is the function $f(z_{1,1}, z_{2,1}) = z_{1,1}z_{2,1}$, then $f(x, y) = x \circ y$. Additionally we can prove that they are homomorphic.

**Lemma 2.** *If $h$ is a PCIS-homomorphism and $f$ is a $k$-ary linear regular function, with $k > 0$, then:*

$$h(f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)) = f(h(\boldsymbol{x}_1), \ldots, h(\boldsymbol{x}_k)).$$

For reasons of space we omit the full proofs; but the proofs require that the functions be non-deleting and non-permuting and require Axioms E and V in the definition of the PCIS. In the case where $k = 0$ and $f$ is variable free this property is trivial.

We can now define the least fixed point semantics of a 2-MCFG in a non-standard way: rather than defining them directly in $\mathfrak{F}(\Sigma)$ which would be completely standard, we will define them in an arbitrary PCIS.

**Definition 6.** *Suppose we have an MCFG $G$ with nonterminals $V$ and a PCIS $A$. Let $\Xi_A^V$ be the set of all functions from nonterminals to elements of $A$; where nonterminals of dimension $d$ are mapped to elements of $A_d$. We define $\bigvee$ and $\leq$ componentwise on $\Xi_A^V$.*

*In particular we define $\xi_\perp$ by $\xi_\perp(N) = \perp_{\dim(N)}$, for all $N \in V$.*

Let $h$ be a homomorphism from $\mathfrak{F}(\Sigma)$ to $A$. Each production $\pi \in P$ in the grammar then defines a function $\Phi^h_\pi$ from $\Xi^V_A \to \Xi^V_A$. If $\pi$ is the $k$-ary production ($k > 0$)

$$N_0 \to f(N_1, \ldots, N_k)$$

then $\Phi^h_\pi(\xi) = \xi'$ is the element of $\Xi^V_A$ which assigns $\perp_d$ to all nonterminals that are not $N_0$, and to the element $N_0$ assigns $f(\xi(N_1), \ldots, \xi(N_k))$.

$$\xi'(N) = \begin{cases} f(\xi(N_1), \ldots, \xi(N_k)) & \text{if } N = N_0, \\ \perp_{\dim(N)} & \text{otherwise.} \end{cases}$$

If $k = 0$ and $\pi$ is of the form $N_0 \to w$, where $w \in (\Sigma^*)^{\dim(N_0)}$, then we use the homomorphism $h$ to define $\xi'$ to be

$$\xi'(N) = \begin{cases} h_{\dim(N)}(\{w\}) & \text{if } N = N_0, \\ \perp_{\dim(N)} & \text{otherwise.} \end{cases}$$

**Definition 7.** *For a grammar $G$, with set of productions $P$ we define a function $\Phi^h_G$ from $\Xi^V_A$ to $\Xi^V_A$.*

$$\Phi^h_G(\xi) = \bigvee \{ \Phi^h_\pi(\xi) \mid \pi \in P \}$$

We define recursively $\Phi^{h,n}_G(\xi) = \Phi^h_G(\Phi^{h,n-1}_G(\xi))$. By Kleene's fixed point theorem, the least fixed point, $\xi^h_G$, of this function exists and is equal to $\bigvee_n \Phi^{h,n}_G(\xi_\perp)$. This least fixed point coincides with the standard derivational semantics of MCFGS when the homomorphism $h$ is the identity: for every nonterminal $N$, $\mathcal{L}(G, N) = \xi_G(N)$. On its own, this result is not very surprising: we have merely defined the derivation process in another way, as has been done before [6] and derived the same result. In order to say something interesting we need to use the homomorphism.

**Lemma 3.** *Suppose we have a homomorphism $h$ from $\mathfrak{F}(\Sigma)$ to some PCIS $A$. Then write $\Phi^h_G$ for the interpretation function in $A$ with $h$. Then for any $\xi \in \Xi^V_{\mathfrak{F}(\Sigma)}$*

$$h(\Phi^\iota_G(\xi)) = \Phi^h_G(h(\xi))\,,$$

*where $\iota$ is the identity.*

The proof is immediate given Lemma 2.

## 5 The Syntactic Concept Lattice

We now extend the syntactic concept lattice [2], from the structure appropriate to CFGS which is a CIS, to the structure appropriate for 2-MCFGS which is a PCIS.

We fix a language $L$ over $\Sigma$. We have a distinguished symbol $\square \notin \Sigma$. A 2-word is an ordered pair of strings $(u, v)$, an element of $\Sigma^* \times \Sigma^*$. A 1-context is a string in $\Sigma^* \square \Sigma^*$ and a 2-context is a string in $\Sigma^* \square \Sigma^* \square \Sigma^*$. See for example [11].

We can combine a $d$-context and a $d$-word using the wrap operation $\odot$. This is defined as $l\Box r \odot u = lur$ and $l\Box m\Box r \odot (u, v) = lumvr$. We extend this to sets of $d$-words and $d$-contexts in the usual way.

**Definition 8.** *We define polar maps: we will use $(\cdot)^{\triangleright}$ for the maps from sets of $d$-words to sets of $d$-contexts and $(\cdot)^{\triangleleft}$ for the maps from sets of $d$-contexts to sets of $d$-words*

*Given a set of 1-words $x$ ($x \in \mathcal{P}(\Sigma^*)$) define*

$$x^{\triangleright} = \{l\Box r \in \Sigma^*\Box\Sigma^* \mid l\Box r \odot x \subseteq L\}\,.$$

*Given a set of 2-words $x$ define*

$$x^{\triangleright} = \{l\Box m\Box r \in \Sigma^*\Box\Sigma^*\Box\Sigma^* \mid l\Box m\Box r \odot x \subseteq L\}\,.$$

*Given a set of 1-contexts $t$ we define*

$$t^{\triangleleft} = \{u \in \Sigma^* \mid t \odot u \subseteq L\}\,.$$

*Given a set of 2-contexts $t$ we define*

$$t^{\triangleleft} = \{(u, v) \in \Sigma^* \times \Sigma^* \mid t \odot (u, v) \subseteq L\}\,.$$

This gives a pair of Galois connections, which as is well known give rise to a lattice [3]. We are interested in the closed sets of $d$-words, which are those sets $x$ such that $x = x^{\triangleright\triangleleft}$. These sets naturally form a PCIS.

**Definition 9.** *The syntactic concept lattice of order 2 of the language $L$, written $\mathfrak{B}(L)$ is the following structure.*

- $A_1 = \mathfrak{B}^1(L) = \{\, x \in \mathcal{P}(\Sigma^*) \mid x = x^{\triangleright\triangleleft} \,\}$
- $A_2 = \mathfrak{B}^2(L) = \{\, x \in \mathcal{P}(\Sigma^* \times \Sigma^*) \mid x = x^{\triangleright\triangleleft} \,\}$
- $\epsilon = \{\lambda\}^{\triangleright\triangleleft}$ *and* $\bot = \emptyset^{\triangleright\triangleleft}$ *(in sort 1).*
- *For* $x, y \in A_1$, $x \circ y = (x \cdot y)^{\triangleright\triangleleft}$ *and* $x \otimes y = (x \times y)^{\triangleright\triangleleft}$
- *For* $X \subseteq A_d$, $\bigvee_d X = (\bigcup X)^{\triangleright\triangleleft}$

**Lemma 4.** *$\mathfrak{B}(L)$ is a* PCIS *and the map from $\mathfrak{F}(\Sigma)$ given by $x \to x^{\triangleright\triangleleft}$ is a* PCIS*-homomorphism.*

*Proof.* (Sketch) Suppose $x, y \in \mathcal{P}(\Sigma^*)$. Observe that $(x \cdot y)^{\triangleright\triangleleft} = (x^{\triangleright\triangleleft} \cdot y^{\triangleright\triangleleft})^{\triangleright\triangleleft}$, and $(x \times y)^{\triangleright\triangleleft} = (x^{\triangleright\triangleleft} \times y^{\triangleright\triangleleft})^{\triangleright\triangleleft}$.

Moreover if $X \subseteq \mathcal{P}(\Sigma^*)$ then $(\bigcup X)^{\triangleright\triangleleft} = (\bigcup\{x^{\triangleright\triangleleft} \mid x \in X\})^{\triangleright\triangleleft}$. □

Note that this will be finite, by an application of the Myhill-Nerode theorem, if and only if the language $L$ is regular. It is easy to see that if $L = \Sigma^*$, then $\mathfrak{B}(L)$ is isomorphic to $\mathbf{1}$; and the only elements are $\Sigma^*$ and $\Sigma^* \times \Sigma^*$.

*Example 1.* $L = a^*b^*$. This is regular and thus $\mathfrak{B}(L)$ is finite. Then $\mathfrak{B}^1(L) = \{\emptyset, \Sigma^*, \{\lambda\}, a^*, b^*, a^*b^*\}$ and $\mathfrak{B}^2(L) = \{\emptyset, \Sigma^* \times \Sigma^*, \{\langle\lambda, \lambda\rangle\}, a^* \times \{\lambda\}, \{\lambda\} \times b^*, a^* \times b^*, a^* \times a^*, b^* \times b^*, a^* \times a^*b^*, a^*b^* \times b^*\}$. Figure 1 contains a diagram showing these lattices.

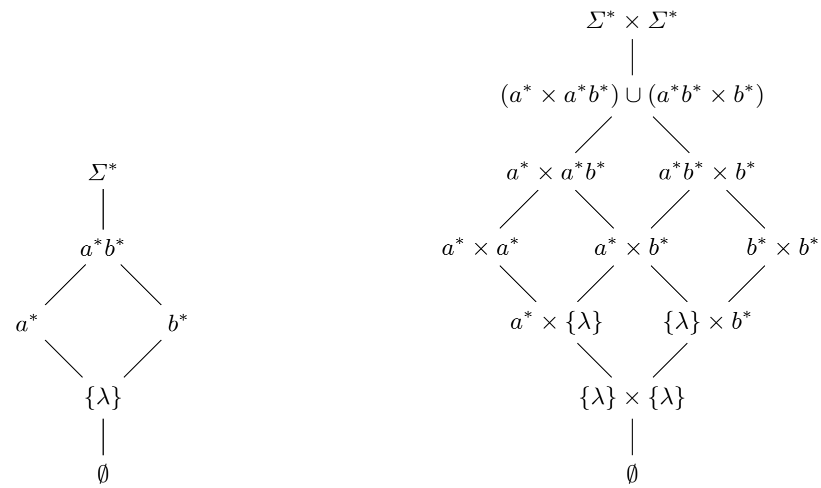


**Fig. 1.** The syntactic concept lattice of order 2 for the regular language $L = a^*b^*$; on the left is a Hasse diagram for $\mathfrak{B}^1(L)$, and on the right for $\mathfrak{B}^2(L)$.

*Example 2.* Suppose $L = \{\, a^m b^n c^m d^n \mid m, n \geq 1 \,\}$. Since this language is not regular $\mathfrak{B}(L)$ has an infinite number of elements. Sort 1 contains $L$, $\Sigma^*$, $\emptyset$, $\{\lambda\}$, as well as $\{a^n bc^n \mid n \geq 1\}$, $\{a^n bbc^n \mid n \geq 1\}$, and many others such as $\{a^m b^{i+j} c^m d^i \mid m \geq 1\}$.

Sort 2 is more complex and contains the infinite sets $\{(a^m, c^m) \mid m \geq 1\}$, $\{(a^{m+k}, c^m) \mid m \geq 1\}$, $\{(a^m, c^{m+k}) \mid m \geq 1\}$ for all $k \geq 1$, and so on, as well as $\{(b^{m+k}, d^m) \mid m \geq 1\}$ and so on. Note that $\mathfrak{B}(L)$ has only countably many elements though $\mathfrak{F}(\Sigma)$ is uncountable.

*Example 3.* Recently Salvati [9] showed that

$$\text{MIX} = \{\, u \in \{a, b, c\}^* \mid |u|_a = |u|_b = |u|_c \,\},$$

where $|u|_a$ denotes the number of occurrences of $a$ in $u$, is a 2-MCFL. Sort 1 ($\mathfrak{B}^1(\text{MIX})$) consists of the sets $x_{i,j} = \{\, u \mid |u|_a - |u|_b = i \text{ and } |u|_a - |u|_c = j \,\}$ for all $(i, j) \in \mathbb{Z} \times \mathbb{Z}$. Sort 2 ($\mathfrak{B}^2(\text{MIX})$) consists of the sets $y_{i,j} = \{\, (u, v) \mid |uv|_a - |uv|_b = i \text{ and } |uv|_a - |uv|_c = j \,\}$ for all $(i, j) \in \mathbb{Z} \times \mathbb{Z}$.

## 6 Homomorphisms and Grammar Morphisms

We will take a standard notion of grammar morphism as applied to MCFGs. Given a function $\phi$ from the set of nonterminals $V_d$ to two disjoint sets $X_d$, we define the image of the grammar $G$, written $\phi(G)$. We extend $\phi$ to a function on productions, mapping

$$\phi(N_0 \to f(N_1, \ldots, N_k)) = \phi(N_0) \to f(\phi(N_1), \ldots, \phi(N_k))$$

and $\phi(N_0 \to w) = \phi(N_0) \to w$. Then $\phi(\langle \Sigma, V, S, P \rangle) = \langle \Sigma, \phi(V), \phi(S), \phi(P) \rangle$.

If $\phi$ is injective then the $G$ is isomorphic to $\phi(G)$. If not, then two nonterminals must have been merged; it is easy to see that in any event $\mathcal{L}(G) \subseteq \mathcal{L}(\phi(G))$.

**Definition 10.** *If $\mathcal{L}(G) = \mathcal{L}(\phi(G))$ then $\phi$ is an* exact *morphism.*

If we have two distinct nonterminals $M, N$ that are merged by an exact morphism $\phi$, (i.e. $\phi(M) = \phi(N)$), then $\phi(G)$ is strictly smaller than $G$ yet defines the same language. In this case we say that $G$ has mergeable nonterminals. Requiring that a grammar does not have mergeable nonterminals is a very weak and natural condition on the grammars that we might want to use in practice or output from a learning algorithm.

Now we come to a crucial point: homomorphisms from $\mathfrak{F}(\Sigma)$ to some other PCIS induce an MCFG-morphism.

**Definition 11.** *Suppose $A$ is a* PCIS *and $h$ a* PCIS*-homomorphism from $\mathfrak{F}(\Sigma)$ to $A$. Then define $\phi_h$ to be a function from $V \to A$ defined by*

$$\phi_h(N) = h(\mathcal{L}(G, N)) \,.$$

This morphism will merge two nonterminals iff their languages are mapped to the same element of $A$ by $h$. Thus we can merge elements *algebraically*. As a special case of this morphism, we have the case where $h$ is the identity. In this case, it merges two nonterminals of dimension $d$ only if they generate exactly the same set of $d$-words. Clearly in this case the language generated by the grammar will be unchanged.

Given that the derivation operations of the grammar are homomorphic with respect to the algebra we can prove the following theorem.

**Theorem 1.** *Suppose we have a* 2-MCFG*, $G$, and a* PCIS *$A$ and a homomorphism $h$ from $\mathfrak{F}(\Sigma) \to A$. Then for all nonterminals $N$ in $G$:*

$$h(\mathcal{L}(\phi_h(G), \phi_h(N)) = h(\mathcal{L}(G, N)).$$

*Proof.* (Sketch) We define a grammar $U$ with the same set of nonterminals as $G$ but with only the following set of unary productions.

$$P_h = \{ N \to \iota(M) \mid h(\mathcal{L}(G, N)) = h(\mathcal{L}(G, M)) \} \,,$$

where $\iota$ is the identity function. Let $H$ be the grammar $G$ with this additional set of productions. Then we can see that $\Phi^\iota_H = \Phi^\iota_G \vee \Phi^\iota_U$. Clearly $\xi^\iota_H \geq \xi^\iota_G$.

- By the definitions we have that $\Phi^h_U(h(\xi^\iota_G)) = h(\xi^\iota_G)$.
- Using Lemma 3, $h(\Phi^\iota_U(\xi_G)) = \Phi^h_U(h(\xi^\iota_G)) = h(\xi^\iota_G)$.
- We can show that if $h(\xi) \leq h(\xi^\iota_G)$ then $h(\Phi^\iota_U(\xi)) \leq h(\xi^\iota_G)$, and furthermore $h(\Phi^\iota_G(\xi)) \leq h(\xi^\iota_G)$ and therefore $h(\Phi^\iota_H(\xi)) \leq h(\xi^\iota_G)$.
- By induction on $n$ we can show that for all $n$, $h(\Phi^{\iota,n}_H(\xi_\perp)) \leq h(\xi^\iota_G)$.
- Finally we can show that $h(\xi^\iota_H) \leq h(\xi^\iota_G)$ which completes the proof. $\square$

Informally, this says the original grammar's derivations are mapped to the same element that the merged grammar's derivations are mapped to. As a consequence, using the fact that $h^*(h(x)) \geq x$ for all $x$, we can easily derive the following algebraic bound on the languages derived in the merged grammar: this lemma unifies these two separate strands. On the left we have a grammar which is the result of a morphism; on the right a purely algebraic bound.

**Lemma 5.** *For any grammar $G$, nonterminal $N$ and homomorphism $h$,*

$$\mathcal{L}(\phi_h(G), N) \subseteq h^*(h(\mathcal{L}(G, N)))\,.$$

We now define the natural extension of the classic automata theoretic notion of a monoid recognizing a regular language [4].

**Definition 12.** *Let $A$ be a* PCIS *and $h$ a* PCIS*-homomorphism from $\mathfrak{F}(\Sigma) \to A$, and $L$ some language over $\Sigma$. Then we say that $A$* recognizes *the language $L$ through $h$ iff*

$$L = h^*(h(L))\,.$$

By Lemma 5, if $G$ generates $L$ and $A$ recognizes $L$ through $h$, then $\phi_h$ is an exact morphism.

Note that if $A$ is the trivial one-element PCIS ($A_1 = \{\bot\}, A_2 = \{\bot \otimes \bot\}$), then $h^*(h(L)) = \Sigma^*$ and so in general, this will not recognize any language apart from $\Sigma^*$. In this case the PCIS is too coarse or small to recognize the language.

Clearly $\mathfrak{F}(\Sigma)$ recognizes the language through the identity homomorphism, and $\mathfrak{B}(L)$ recognizes $L$ through the homomorphism $X \to X^{\rhd\lhd}$, since $L = L^{\rhd\lhd}$.

Our main theorem is then immediate.

**Theorem 2.** *Let $G$ be a 2-*MCFG *that defines a language $L$, and define $\phi_L$ to be the morphism given by $\phi_L(N) = \mathcal{L}(G, N)^{\rhd\lhd}$. Then $\phi_L$ is an exact morphism: $\mathcal{L}(\phi_L(G)) = L$.*

We will now show that $\mathfrak{B}(L)$ is the smallest PCIS that recognizes the language.

**Lemma 6.** *Suppose, $A$ is some* PCIS *that recognizes $L$ through a morphism $h$. Suppose $x, y \in \mathcal{P}((\Sigma^*)^d)$ are such that $h_d(x) = h_d(y)$; then $x^{\rhd} = y^{\rhd}$.*

*Proof.* Suppose $l\Box r \in x^{\rhd}$ for some $l, r \in \Sigma^*$. That means that $lxr \subseteq L$; so $h(lxr) \leq h(L)$ since $h$ is monotonic. This implies that $h(\{l\}) \circ h(x) \circ h(\{r\}) \leq h(L)$ using the fact that $h$ is a homomorphism. So $h(\{l\}) \circ h(y) \circ h(\{r\}) \leq h(L)$ since $h(x) = h(y)$. Therefore $h(\{l\}y\{r\}) \leq h(L)$ since it is a homomorphism. So $h^*(h(\{l\}y\{r\})) \leq h^*(h(L))$ since $h^*$ is monotonic. Now $h^*(h(L)) = L$ since $A$ recognizes $L$. Therefore $h^*(h(\{l\}y\{r\})) \leq L$. So using the fact that $h^*(h(x)) \geq x$ we have $\{l\}y\{r\} \leq h^*(h(\{l\}y\{r\})) \leq h^*(h(L)) = L$. So $l\Box r \in y^{\rhd}$. The converse argument is identical and so $x^{\rhd} = y^{\rhd}$.

The argument for sort 2 is identical but requires some additional notation so for reasons of space we omit it. $\square$

We now show that $\mathfrak{B}(L)$ is the unique up to isomorphism smallest structure that recognizes the language: a terminal object in category theoretic terms.

**Theorem 3.** *Suppose $A$ is a* PCIS *that recognizes $L$ through a surjective homomorphism $j$. Let $h_L$ be the homomorphism $\mathfrak{F}(\Sigma) \rightarrow \mathfrak{B}(L)$ given by $h_L(x) = x^{\triangleright\triangleleft}$. Then there is a homomorphism $f$ from $A$ to $\mathfrak{B}(L)$ such that $h_L(x) = f(j(x))$.*

*Proof.* (Sketch) We define $f$ as the pair of functions $f_d : A_d \rightarrow \mathfrak{B}^d(L)$

$$f_d(x) = j_d^*(x)^{\triangleright\triangleleft}$$

By Lemma 6 for any $x, y \in (\mathcal{P}(\Sigma^*))^d$, if $j_d(x) = j_d(y)$ then $x^{\triangleright} = y^{\triangleright}$. This means that $j_d^*(j_d(x)) \subseteq x^{\triangleright\triangleleft}$, and therefore $(j_d^*(j_d(x)))^{\triangleright\triangleleft} = x^{\triangleright\triangleleft}$; in other words $h_L(x) = f(j(x))$. We can then verify that it is a homomorphism. □

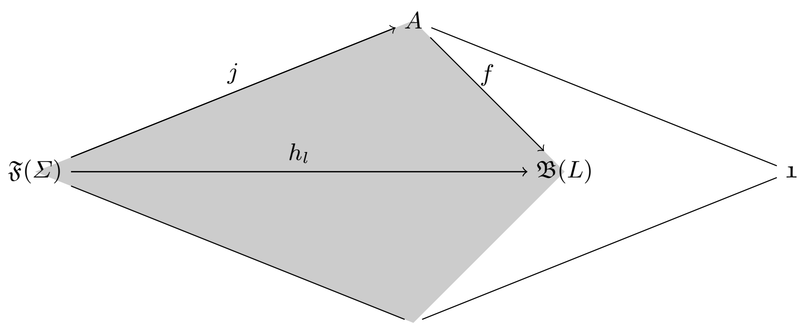


**Fig. 2.** A diagram illustrating the universal property of the lattice described in Theorem 3. The shaded area consists of those PCIS which recognize $L$; the unshaded area contains those which do not; on the far right is the trivial single element PCIS, $\mathbf{1}$. If $A$ recognizes the language there is a unique homomorphism from $A$ to $\mathfrak{B}(L)$.

Theorem 2 means that any grammar for $L$ without mergeable nonterminals will have nonterminals that correspond to elements of $\mathfrak{B}(L)$. Theorem 3 shows that $\mathfrak{B}(L)$ is the smallest structure that has this property.

## 7 Conclusion

The first important observation is that the set of derivation contexts of nonterminal of dimension 2 in an MCFG correspond exactly to the string contexts of the form $l\Box m\Box r$. The second observation is that if the family of operations that build up objects bottom up are viewed algebraically, then homomorphisms of an appropriate algebra will induce "nice" morphisms of the grammar, and typically allow a precise characterization of some minimal structure. This general strategy is applicable to a broader range of grammatical formalisms. The extension to MCFGs of higher dimension is we think straightforward; but we feel that this does not exhaust the possibilities of this family of techniques.

From one perspective, this is very natural: each nonterminal in a grammar with context-free derivations defines a decomposition into the set of derivation contexts and the set of yields. The result here shows that we can reverse this process and define nonterminals based on decompositions: the "minimal" grammar will then be based on maximal decompositions.

The results we have presented here imply that we can assume, without loss of generality, that the nonterminals of an MCFG correspond to, or represent, elements of the syntactic concept lattice. Rather than being arbitrary atomic symbols, they have a rich algebraic structure. This implies that it is possible to develop techniques for example for translating between CFGs and MCFGs, for minimizing grammars and so on. This is also an important step towards the structural learning of MCFGs.

## References


1. Alexander Clark. Learning context free grammars with the syntactic concept lattice. In José Sempere and Pedro García, editors, *Grammatical Inference: Theoretical Results and Applications. Proceedings of the International Colloquium on Grammatical Inference*, pages 38–51. Springer, 2010.
2. Alexander Clark. The syntactic concept lattice: Another algebraic theory of the context-free languages? *Journal of Logic and Computation*, 2013.
3. B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 2002.
4. S. Eilenberg. *Automata, languages, and machines*. Academic press, 1974.
5. S. Ginsburg and H.G. Rice. Two families of languages related to ALGOL. *Journal of the ACM (JACM)*, 9(3):350–371, 1962.
6. Annius Groenink. *Surface without structure: Word Order and Tractability Issues*. PhD thesis, University of Utrecht, 1997.
7. G. Morrill. *Categorial grammar: Logical syntax, semantics and processing*. Oxford University Press, 2011.
8. F.C.N. Pereira and S.M. Shieber. The semantics of grammar formalisms seen as computer languages. In *Proceedings of the 10th International Conference on Computational Linguistics and 22nd annual meeting on Association for Computational Linguistics*, pages 123–129. Association for Computational Linguistics, 1984.
9. Sylvain Salvati. MIX is a 2-MCFL and the word problem in $\mathbb{Z}^2$ is solved by a third-order collapsible pushdown automaton. Technical Report Inria-00564552, version 1, INRIA, 2011.
10. Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. On multiple context-free grammars. *Theoretical Computer Science*, 88(2):191–229, 1991.
11. R. Yoshinaka. Efficient learning of multiple context-free languages with multidimensional substitutability from positive data. *Theoretical Computer Science*, 412(19):1821 – 1831, 2011.
12. Ryo Yoshinaka. Integration of the dual approaches in the distributional learning of context-free grammars. In Adrian Horia Dediu and Carlos Martín-Vide, editors, *LATA*, volume 7183 of *Lecture Notes in Computer Science*, pages 538–550. Springer, 2012.