A Hierarchy of Languages with Catenation and Shuffle

Nils Erik Flick and Manfred Kudlek

Fachbereich Informatik, MIN-Fakultät, Universität Hamburg, DE
email: {flick,kudlek}@informatik.uni-hamburg.de

Abstract. We present basic structures, definitions, normal forms, and a hierarchy of languages based on catenation, shuffle and their iterations, defined by algebraic closure or least fix point solution of systems of equations.

Keywords: shuffle catenation languages, hierarchy, algebraic characterization

1 Introduction

In this paper, we establish a hierarchy of languages expressing possibilities of iterated sequential and parallel compositions of basic events, based on extending the construction principles behind the well-known regular and context-free languages with another operation known as the shuffle [8, 9]. Related investigations, in particular on shuffle languages, are given in [1, 5, 6]. There only certain combinations of catenation, shuffle and their iterations have been considered. Such combinations of both operators are especially useful for modelling some areas of concurrency, and in particular the behaviour of client/server systems [2], and also for semantics of Petri nets, such as interleaving semantics.

In section 2 we introduce or recall the basic definitions and structures needed for further investigation, such as monoids, semirings, bi-monoids and bi-semirings, furthermore systems of equations and their least fix point solutions, and normal forms for them, as well as algebraic closure of finite sets under certain language operators. In section 3 we investigate the complete hierarchy of language classes defined as algebraic closures of union, catenation, shuffle and their iterations applied on the class of finite languages, as well as classes defined by least fix point solutions of systems of equations, and their relation to the Chomsky hierarchy. Section 4 offers an outlook for further research in the area such as closure of language classes under certain operators, or decidability problems.

2 Definitions and Basic Structures

Formal language theory normally deals with subsets of Σ^*, all words over a finite alphabet, using as basic binary operator catenation, denoted by $\odot$ in the sequel. This gives a basic monoid with $\odot$ and λ, the empty word. Other binary operators

have also been considered, as e.g. shuffle, denoted by $\sqcup\!\sqcup$, which we define below under "basic structures".

In contrast to catenation $\sqcup\!\sqcup$ is also commutative; like catenation, it can be used to define another monoid with the finite subsets of Σ^* as domain. Another possibility is to combine both operators. Extending $\odot$ and the domain of $\sqcup\!\sqcup$ to languages gives rise to a basic bi-monoid with $\{\lambda\}$ as common neutral element, and the class of finite subsets of Σ^* as domain.

2.1 Basic Structures

In what follows we present, partially recalling, the definitions of such structures more precisely. For $w \in \Sigma^*$ let $||w||$ denote the *length* of w. If $A \subseteq \Sigma^*$ then $|A|$ denotes the *cardinality* of A which also can be infinite. In particular, $||\lambda|| = 0$. For $A \subseteq \Sigma^*$ let $||A|| = max\{||w|| \mid w \in A\}$ the *norm* of A.

Based on the monoid $\mathcal{M}_\odot = (\Sigma^*; \lambda, \odot)$, where Σ is a (finite) alphabet, $\odot$ the binary operation *catenation*, and λ is the neutral element for the operator $\odot$, $\mathcal{S}_\odot = (2^{\Sigma^*}; \emptyset, \{\lambda\}, \cup, \odot)$ is an ω-complete semiring since $\odot$ is associative, $\cup$ is commutative, associative and distributive with $\odot$, and

$$A \odot \bigcup_j B_j = \bigcup_j (A \odot B_j) \ , \ (\bigcup_j B_j) \odot A = \bigcup_j (B_j \odot A)$$

for all $A, B_j \in 2^{\Sigma^*}$ where the union can also be infinite. Elements (words) $w \in \Sigma^*$ or singletons $\{w\}$ can be seen as basic elements (atoms). At a somehow higher level also finite sets can serve as such. For that let $2_1^{\Sigma^*} = \{\alpha \in 2^{\Sigma^*} \mid |\alpha| = 1\}$,the class of singletons, and consider $\mathcal{FIN} = 2_f^{\Sigma^*} = \{\alpha \in 2^{\Sigma^*} \mid |\alpha| < \infty\}$, the class of finite sets of words. This can be generated from the first one by usual extension to sets. For a general treatment of semirings and related structures see [12].

A similar construct holds for the *shuffle* operator $\sqcup\!\sqcup : \Sigma^* \times \Sigma^* \to 2^{\Sigma^*}$, which can be defined recursively as follows: For all $a, b \in \Sigma$ and $v, w \in \Sigma^*$, $\lambda \sqcup\!\sqcup w = w \sqcup\!\sqcup \lambda = \{w\}$; $aw \sqcup\!\sqcup bv = \{a\} \odot (w \sqcup\!\sqcup bv) \cup \{b\} \odot (aw \sqcup\!\sqcup v)$ and extended to sets from now on: $\sqcup\!\sqcup : 2^{\Sigma^*} \times 2^{\Sigma^*} \to 2^{\Sigma^*}$, $A \sqcup\!\sqcup B = \bigcup_{w \in A, v \in B} w \sqcup\!\sqcup v$.

Here the basic monoid is $\mathcal{M}_{\sqcup\!\sqcup} = (2_f^{\Sigma^*}; \{\lambda\}, \sqcup\!\sqcup)$. Then $\mathcal{S}_{\sqcup\!\sqcup} = (2^{\Sigma^*}; \emptyset, \{\lambda\}, \cup, \sqcup\!\sqcup)$ is an ω-complete semiring as well since $\sqcup\!\sqcup$ is associative, $\cup$ is commutative, associative and distributive with $\sqcup\!\sqcup$, and

$$A \sqcup\!\sqcup \bigcup_j B_j = \bigcup_j (A \sqcup\!\sqcup B_j) \ , \ (\bigcup_j B_j) \sqcup\!\sqcup A = \bigcup_j (B_j \sqcup\!\sqcup A)$$

for all $A, B_j \in 2^{\Sigma^*}$.

A *bi-monoid* is a structure $\mathcal{D} = (\mathbf{D}; 1, \odot, \otimes)$ where $\odot$ and $\otimes$ are associative binary operations on $\mathbf{D}$, and 1 is the common neutral element.

A *bi-semiring* is a structure $\mathcal{B} = (\mathbf{B}; 0, 1, \oplus, \odot, \otimes)$ where $(\mathbf{B}; 0, 1, \oplus, \odot)$ and $(\mathbf{B}; 0, 1, \oplus, \otimes)$ are semirings, i.e. $\oplus$ is a commutative and associative binary operation on $\mathbf{B}$, $\odot$ and $\otimes$ are associative binary operations on $\mathbf{B}$, 0 is the neu-

tral element of $\oplus$, and 1 the common neutral element of $\odot$ and $\otimes$. Furthermore, $\oplus$ is distributive with $\odot$ and $\otimes$, i.e. $x \oplus (y \odot z) = (x \odot y) \oplus (x \odot z)$ and $x \oplus (y \otimes z) = (x \otimes y) \oplus (x \otimes z)$ for all $x, y, z \in \mathbf{B}$.

A bi-semiring $\mathcal{B}$ is *ω-complete* if

$$x \odot \bigoplus_j y_j = \bigoplus_j (x \odot y_j) \ , \ (\bigoplus_j y_j) \odot x = \bigoplus_j (y_j \odot x) \ ,$$

$$x \otimes \bigoplus_j y_j = \bigoplus_j (x \otimes y_j) \ , \ (\bigoplus_j y_j) \otimes x = \bigoplus_j (y_j \otimes x)$$

for $x, y_j \in \mathbf{B}$ and arbitrary (finite or infinite) 'sums' $\bigoplus$ with $\oplus$.

Let Σ be an alphabet. Then $\mathcal{D}_{\odot \sqcup} = (2_1^{\Sigma^*}; \{\lambda\}, \odot, \sqcup)$ is the basic bi-monoid for formal languages using both operations, $\odot$ and $\sqcup$, similar to $\mathcal{B}_{\odot}$ for formal languages with $\odot$ only.

Then $\mathcal{B}_{\odot \sqcup} = (2^{\Sigma^*}; \emptyset, \{\lambda\}, \cup, \odot, \sqcup)$ is a bi-semiring since $\cup$ distributes with $\odot$ and $\sqcup$. This bi-semiring is also ω-complete.

2.2 Systems of Equations

It is well known that the class $\mathcal{CF}$ of context-free languages can be characterized by least fix point solutions of systems of equations using structures based on catenation $\odot$. Similar characterizations can be defined for languages based on structures with $\sqcup$ or with both, $\odot$ and $\sqcup$, respectively.

Let $\mathcal{V}$ be a finite set of *variables*, standing for subsets $X \subseteq \Sigma^*$, and $\mathcal{C}$ a finite set of *constants* $\alpha \in 2_1^{\Sigma^*}$ or $\alpha \in 2_f^{\Sigma^*}$, thus elements of the basic structure. Thus $\mathcal{V} = \{X_1, \cdots, X_m\}$ and $\mathcal{C} = \{\alpha_1, \cdots, \alpha_n\}$.

A *monomial* is a finite expression $m(\bar{X})$ on $\mathcal{V} \cup \mathcal{C}$ using binary operations $\odot$, or $\sqcup$, or both $\odot$ and $\sqcup$, where $\bar{X}$ denotes the tuple of (ordered) variables, e.g. $(X_1 \odot \alpha_1) \sqcup (X_2 \odot X_3)$.

A *polynomial* $p(\bar{X})$ is a finite union of monomials.

A *system of equations* is a system $X_i = p_i(\bar{X})$ $(1 \leq i \leq m)$, or in compact form $\bar{X} = \bar{p}(\bar{X})$.

A system of equations is called *algebraic* if the monomials occurring in the system of equations are arbitrary, *linear* if all monomials have one of the forms $(A \circ X) \circ B$, $A \circ (X \circ B)$, or A with $X \in \mathcal{V}$, A, B expressions of constants only, and $\circ \in \{\odot, \sqcup\}$, *rational* if all monomials have the form $X \circ A$ or A.

If the underlying semiring or bi-semiring is ω-complete such a system has a solution as least fix point. This can be constructed by iteration, starting with $\bar{X}^{(0)} = \bar{\emptyset}$, and iterating $\bar{X}^{(j+1)} = \bar{p}(\bar{X}^{(j)})$.

Clearly, $\bar{X}^{(j)} \subseteq \bar{X}^{(j+1)}$ for $0 \leq j$, where $\subseteq$ is meant componentwise for all $0 \leq i \leq m$, thus a monotone sequence. This is shown by induction, the basis $\bar{\emptyset} \subseteq \bar{X}^{(1)}$ being trivial, and with induction hypothesis $\bar{X}^{(j)} \subseteq \bar{X}^{(j+1)}$ and $\bar{X}^{(j+1)} = \bar{p}(\bar{X}^{(j)}) \subseteq \bar{p}(\bar{X}^{(j+1)}) = \bar{X}^{(j+2)}$.

Since $\bar{X}^{(j)} \subseteq \bar{\Sigma}^*$, a total upper bound, there exists $lim_{j\to\infty}\bar{X}^{(j)} = \bar{Y}$ which is the least fix point solution, i.e. $\bar{Y} = \bar{p}(\bar{Y})$.

Each component of the least fix point solution defines a formal language of the system's kind (algebraic, linear, rational), e.g the component belonging to the first variable X_1.

Corresponding to the underlying semirings, different language classes can be defined. These are
$\mathcal{ALG}(\odot) = \mathcal{CF}$, $\mathcal{LIN}(\odot) = \mathcal{LIN}$, $\mathcal{RAT}(\odot) = \mathcal{REG}$,
$\mathcal{ALG}(\sqcup) = \mathcal{LIN}(\sqcup) = \mathcal{RAT}(\sqcup) = \mathcal{SHUF}$,
$\mathcal{ALG}(\odot,\sqcup)$, $\mathcal{LIN}(\odot,\sqcup)$, $\mathcal{RAT}(\odot,\sqcup)$.

Example 1. An example of a rational system of equations is given by $X = X \odot \alpha \cup X \sqcup \beta \cup \gamma$ with singletons $\{\alpha, \beta, \gamma\} = \mathcal{C}$ from disjoint alphabets. The least fix point solution is a set of languages defined by the following terms in prefix notation, where $h : \{\odot,\sqcup\}^* \to \mathcal{C}$ is a homomorphism defined by $h(\odot) = \alpha$, $h(\sqcup) = \beta$, and ^{R} denotes the reverse.

$$\bigcup_{u\in\{\odot,\sqcup\}} u\gamma h(u)^R \;=\; (\gamma \odot \alpha^{\odot}) \sqcup \beta^{\sqcup}$$

In case of one single commutative operator the classes of algebraic, linear, and rational languages coincide [12], such as e.g for $\sqcup$.

Whereas in a system of equations one gets least fix points solutions for all variables, grammars just produce the solution of a distinguished variable, the *initial* variable. The equations are written as basic derivation steps, e.g. for
$X = \alpha \odot (Y \odot \beta) \cup (Y \sqcup Z) \odot X \cup \gamma$ gives the *productions* $X \to \alpha \odot (Y \odot \beta)$, $X \to \alpha \odot (Y \odot \beta)$, $X \to \gamma$.

2.3 Algebraic Closures

Another characterization of languages is achieved by application of some language operators on a basic class of languages. The operators can be applied either in arbitrary, prescribed, or somehow mixed order. In this way one gets *algebraic closures* under the language operators, i.e the least class of languages closed under such operators.

Here we consider the operators $\cup$, $\odot$, and $\sqcup$, as well as their iterations $^{\odot}$ and $^{\sqcup}$, applied on members of the basic class $\mathcal{FIN}$ of finite languages. Using the operator $\cup$ one could also take Σ_1^*, the class of singletons as basic class. But for convenience we start with $\mathcal{FIN}$.

$(\cup, \odot, ^{\sqcup})(\mathcal{FIN})$ means that the operators $\cup$, $\odot$, and $\sqcup$ are applied in arbitrary order and arbitrary often on finite sets, whereas $(\cup, ^{\sqcup})(\odot)(\sqcup)(\mathcal{FIN})$ means that at first $\sqcup$ is applied arbitrary often, then $\odot$ arbitrary often, and finally $\cup$ and $^{\sqcup}$ arbitrary often and in arbitrary order. Note the non-application of corresponding operators is always understood, i.e. each algebraic closure also contains $\mathcal{FIN}$.

Note that for any set A the following facts hold:
$(A^{\sqcup})^{\odot} = (A^{\odot})^{\sqcup} = (A^{\sqcup})^{\sqcup} = A^{\sqcup}$, $A^{\odot} \subseteq A^{\sqcup}$, and $(A^{\odot})^{\odot} = A^{\odot}$.

Important classes are $(\cup,\sqcup,^{\sqcup})(\mathcal{FIN}) = \mathcal{SHUF}$, $(\cup,\odot,^{\odot},^{\sqcup})(\mathcal{FIN}) = \mathcal{ER}$, $(\cup,\sqcup,^{\odot},^{\sqcup})(\mathcal{FIN}) = \mathcal{ES}$, and $(\cup,\odot,\sqcup,^{\odot},^{\sqcup})(\mathcal{FIN}) = \mathcal{SE}$ where $\mathcal{ES}$ stands for *extended shuffle expression*, analogous to $\mathcal{ER}$ for *extended regular expression*.

2.4 Normal Forms

For any system of equations an equivalent one with possibly more variables can be constructed such that the solution of the new system coincides with the solution of the old system on the variables of the old system, and the monomials of the new system are of a simple form. This will be shown for systems with operators $\odot,\sqcup$. Since $\cup$ is idempotent, i.e. $e \cup e = e$ for any expression, wlog e occurs only once as a monomial in a polynomial.

In case of an algebraic system consider a monomial. If it is of the form $Y \circ Z$, Y, or α, where $Y, Z \in \mathcal{V}$, $\circ \in \{\odot,\sqcup\}$, $\alpha \in \mathcal{C}$, leave it unchanged. If the form is $Y \circ \alpha$ or $\alpha \circ Y$, add a new variable Z, replace the monomial by $Y \circ Z$ or $Z \circ Y$, respectively, and add a new equation $Z = \alpha$ to the system.

If it has the form $e_1 \circ e_2$ where e_1, e_2 are expressions such that it is not of the form above, add new variables Z_1, Z_2, replace $e_1 \circ e_2$ by $Z_1 \circ Z_2$ and add new equations $Z_1 = e_1$, $Z_2 = e_2$ to the system. Repeat the procedure until all (also new) monomials are of the form above.

Note that also expressions A consisting only of constants are reduced.

Thus only forms $Y \circ Z$, Y, or α are achieved.

In case of a linear system of equations leave unchanged monomials of the form Y, $Y \circ \alpha$, $\alpha \circ Y$, and α. Otherwise proceed as for algebraic systems.

Thus there are only monomials of the form Y, $\alpha \circ Y$, $Y \circ \alpha$, or α.

In case of a rational system of equations leave unchanged monomials of the form Y, $Y \circ \alpha$, or α. Otherwise proceed as for algebraic systems. Only expressions of constants are processed. Thus one gets the normal form Y, $Y \circ \alpha$, or α.

Another reduction is the eliminations of polynomials of the form Y.

If X occurs in its own polynomial then it can be removed. To show that let $X = p_X(\bar{X}) = q_X(\bar{X}) \cup X$ be the component for X in the system of equations $\bar{X} = \bar{p}(\bar{X})$. Consider also the system $\bar{X}' = \bar{p}'(\bar{X}')$ which is identical to the first one except for $p_X(\bar{X})$ replaced by $q_X(\bar{X}')$. Then $\bar{X}^{(j)} = \bar{X}'^{(j)}$ for all $j \geq 0$ in the fix point approximation. This is shown by induction.

$\bar{X}^{(0)} = \bar{\emptyset} = \bar{X}'^{(0)}$

With the induction hypothesis $\bar{X}^{(j)} = \bar{X}'^{(j)}$ follows

$\bar{X}^{(j+1)} = \bar{p}(\bar{X}^{(j)}) = \bar{p}(\bar{X}'^{(j)})$

where $p_X(\bar{X}^{(j)}) = q_X(\bar{X}^{(j)}) \cup \bar{X}^{(j)} = q_X(\bar{X}'^{(j)}) \cup \bar{X}'^{(j)} = q_X(\bar{X}'^{(j)})$

since $\bar{X}'^{(j)} \subseteq \bar{p}(\bar{X}'^{(j)}) = \bar{X}'^{(j+1)}$,

and therefore $\bar{p}(\bar{X}'^{(j)}) = \bar{p}'(\bar{X}'^{(j)}) = \bar{X}'^{(j+1)}$, yielding $\bar{X}^{(j+1)} = \bar{X}'^{(j+1)}$.

Hence both systems have the same least fix point solution.

Therefore:

Lemma 1. *To any system of equations there exists an equivalent one with respect to least fixpoint, with following normal forms of the monomials:*

algebraic: $Y \odot Z$, $Y \sqcup Z$, α
linear: $Y \odot \alpha$, $Y \sqcup \alpha$, $\alpha \odot Y$, $\alpha \sqcup Y$, α
rational: $Y \odot \alpha$, $Y \sqcup \alpha$, α
where $Y, Z \in \mathcal{V}$, $\alpha \in 2_1^{\Sigma^*}$.

3 Hierarchies

In this section we present two language hierarchies, a *lower* and an *upper* one.

3.1 The Lower Hierarchy

The first hierarchy we will establish is one of families of languages (in the sense of [8]) which are obtained as the closure of the family of finite languages under some of the operations $\cup$, $\odot$, $\sqcup$, $^{\odot}$, $^{\sqcup\!\sqcup}$, extended in the obvious way to families of languages. It is shown in Figure 1 (with $(\mathcal{FIN})$ understood). By Lemma 11 in subsection 3.2, all of these are subsets of $RAT(\odot, \sqcup)$.
Two classes coincide since $\mathcal{REG}$ is closed under $\sqcup$ [3] which we recall here:

Proposition 1. $(\cup, \odot, \sqcup, ^{\odot})(\mathcal{FIN}) \subseteq (\cup, \odot, ^{\odot})(\mathcal{FIN})$, *i.e.* $\mathcal{REG}$ *is closed under* $\sqcup$.

Proof. Let $R_1, R_2 \in \mathcal{REG}$. Then R_1, R_2 are accepted by deterministic finite automata $A_1 = (Q_1, \Sigma_1, \delta_1, q_{01}, Q_{f1})$ and $A_2 = (Q_2, \Sigma_2, \delta_2, q_{02}, Q_{f2})$. Construct a NFA $A = (Q, \Sigma, \delta, q_0, Q_f)$ by $Q = Q_1 \times Q_2$, $\Sigma = \Sigma_1 \cup \Sigma_2$, $q_0 = (q_{01}, q_{02})$, $Q_f = Q_{f1} \times Q_{f2}$, $((q, s), x, (q', s)) \in \delta$ if $\delta_1(q, x) = q'$ or $((q, s), y, (q, s')) \in \delta$ if $\delta_2(s, y) = s'$. Then A accepts the language $R_1 \sqcup R_2$. □

From this follows

Theorem 1. $(\cup, \odot, \sqcup, ^{\odot})(\mathcal{FIN}) = (\cup, \odot, ^{\odot})(\mathcal{FIN}) = \mathcal{REG}$.

All of the inclusions in the diagram of Figure 1 are proper; to show this, it is sufficient to prove the following lemmata (by counterexamples):

- $(\odot, ^{\odot})(\mathcal{FIN}) \cap (\odot, ^{\sqcup\!\sqcup})(\mathcal{FIN}) \not\subseteq \mathcal{ES}$ (Lemma 2)
- $(\cup, ^{\odot})(\mathcal{FIN}) \cap (\cup, ^{\sqcup\!\sqcup})(\mathcal{FIN}) \not\subseteq (\odot, \sqcup, ^{\odot}, ^{\sqcup\!\sqcup})(\mathcal{FIN})$ (Lemma 4)
- $(\sqcup, ^{\sqcup\!\sqcup})(\mathcal{FIN}) \not\subseteq \mathcal{ER}$ (Lemma 8)
- $(\sqcup, ^{\odot})(\mathcal{FIN}) \not\subseteq (\cup, ^{\odot}, ^{\sqcup\!\sqcup})(\mathcal{FIN})$ (Lemma 6)
- $(\sqcup, ^{\odot})(\mathcal{FIN}) \not\subseteq (\odot, ^{\odot}, ^{\sqcup\!\sqcup})(\mathcal{FIN})$ (Lemma 5)
- $(^{\sqcup\!\sqcup})(\mathcal{FIN}) \not\subseteq (\cup, \odot, \sqcup, ^{\odot})(\mathcal{FIN})$ (Lemma 10)
- $(^{\odot})(\mathcal{FIN}) \not\subseteq (\cup, \odot, \sqcup, ^{\sqcup\!\sqcup})(\mathcal{FIN})$ (Lemma 9)

Lemma 2. $\{a\}^{\sqcup\!\sqcup} \odot \{b\}^{\sqcup\!\sqcup} = \{a\}^{\odot} \odot \{b\}^{\odot} \notin \mathcal{ES}$.

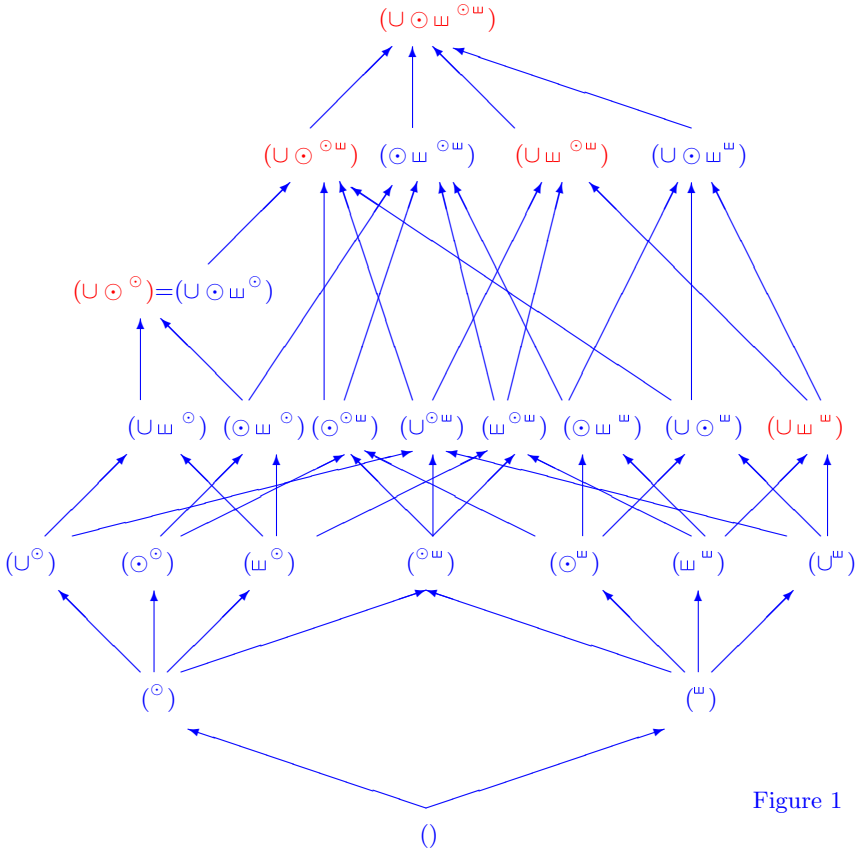

Figure 1

Proof. $\forall L \in \mathcal{ES}\ \exists m \in \mathbb{N} : ((k > 1, \ell > 1, k + \ell > m, a^k b^\ell \in L) \Rightarrow \exists ubav \in L)$. But this is not true for $\{a\}^{\odot} \odot \{b\}^{\odot}$.

Proof by structural induction: A language in $\mathcal{ES}$ $A \cup B$, $A \sqcup B$, $A^{\odot}$ or $A^{\sqcup}$ with $A, B \in \mathcal{L}$. For any finite language L let $m = \|L\|$. If A does not contain infinitely many words of the shape $a^m b^n$ but $A^{\odot}$ or $A^{\sqcup}$ does, then this is due to A containing $a^k b^\ell$ words up to a certain length or a^k and b^ℓ words. In any case, wrongly ordered words result. If A and B have the property, then $A \cup B$ obviously has it and for $A \sqcup B$, a contradiction is also reached if one of them, wlog A, contains a word uav and B contains a word $u_2 b v_2$. □

Lemma 3. *Let $\psi : L \to \mathbb{N}^\Sigma$ be the Parikh mapping that takes a word w to the vector $\psi(w) \in \mathbb{N}^\Sigma$ with components identical to the multiplicities of symbols from Σ, extended to languages.*

For any language $L \in (\odot, \sqcup, ^{\odot}, ^{\sqcup})(\mathcal{FIN})$, we have that

$$(\exists w \in L\ \exists \xi \in \mathbb{N}^\Sigma\ \forall k \in \mathbb{N}\ :\ \psi(w) + k \cdot \xi \in \psi(L))$$
$$\Rightarrow (\forall w \in L\ \exists \xi' \geq \xi\ \forall k \in \mathbb{N}\ :\ \psi(w) + k \cdot \xi' \in \psi(L))\ .$$

Proof. By structural induction over an $(\odot, \sqcup, {}^{\odot}, {}^{\text{ш}})$-term for L. If A, B have the property with vector $\xi_A(w)$ depending on w, resp. $\xi_B(w)$, then in $A \odot B$, the same vectors can be used ($\xi_A(u)$ is good for any word $uv \in A \odot B$ and by assumption some $\xi' \geq \xi_A(u)$ is good for any other word $u_2 v \in A \odot B$ and vice versa). For $A \sqcup B$, the same holds. For $A^{\odot}$ and $A^{\text{ш}}$, any new word can be suffixed with formerly possible iterations. □

Lemma 4. $\{a\}^{\text{ш}} \cup \{b\}^{\text{ш}} = \{a\}^{\odot} \cup \{b\}^{\odot} \notin (\odot, \sqcup, {}^{\odot}, {}^{\text{ш}})(\mathcal{FIN})$.

Proof. Applying Lemma 3 to $w = \{a\}$ and $\xi = \{(b, 1)\}$. □

Lemma 5. $(\sqcup, {}^{\odot})(\mathcal{FIN}) \not\subseteq (\odot, {}^{\odot}, {}^{\text{ш}})(\mathcal{FIN})$.

Proof. Consider $L = \{ab\}^{\odot} \sqcup \{cd, ef\} \in (\sqcup, {}^{\odot})(\mathcal{FIN})$. Assume $L \in (\odot, {}^{\odot}, {}^{\text{ш}})(\mathcal{FIN})$.

Let $L = A^{\odot}$ with $A \neq \emptyset$, $A \neq \{\lambda\}$. Now $cd \in L$. Either $cd \in A^i$ for some i yielding $cdcd \in L$, a contradiction, or $c \in A^i$, $d \in A^j$ for some i, j yielding $dc \in L$, also a contradiction. $L = A^{\text{ш}}$ gives the same contradiction.

Let $L = A \odot B$ with $A \neq \{\lambda\}$, $B \neq \{\lambda\}$.

If there exists $xcy \in A$ then there exists neither $ucv \in B$ nor $u'ev' \in B$ nor $u''fv'' \in B$ since otherwise $xcyuev \in L$ or $xcvu'ev' \in L$ or $xcyu''fv'' \in L$, all contradictions. Therefore either $xcydz \in A$ or $udv \in B$ possible. But then there would be no $xeyfz \in L$, a contradiction. □

Lemma 6. $(\sqcup, {}^{\odot})(\mathcal{FIN}) \not\subseteq (\cup, {}^{\odot}, {}^{\text{ш}})(\mathcal{FIN})$.

Proof. Any language of $\mathcal{L}_2 = (\cup, {}^{\odot}, {}^{\text{ш}})(\mathcal{FIN})$ is a finite union $\bigcup_i L_i$ of languages which are either finite or $L_i = A_i^{\odot}$ or $A_i^{\text{ш}}$ for languages A_i. If $w \in A$, $ww \in A^{\odot}$ and $ww \in A^{\text{ш}}$. Suppose $L_1 = \{a\} \sqcup \{b\}^{\odot} \in \mathcal{L}_2$. Then L_1 is infinite and $a \in L_1$, and a word a^n with $n >$ the longest word in any of the finite languages, must be in one of the languages A_i so we conclude $aa \in L_1$, which is wrong. □

Lemma 7. *Every language $L \in \mathcal{ER}$ can be written as a union as follows, with I a finite set and all $K(i) \in \mathbb{N}$, for a finite number of sets $A_{ik} \in \mathcal{ER}$ which are all either finite or $C_{ik}^{\odot}$ or $C_{ik}^{\text{ш}}$ for some $C_{ik} \in \mathcal{ER}$.*

$$L = \bigcup_{i \in I} \bigodot_{k=0}^{K(i)} A_{ik}$$

This follows from the distributivity of $\odot$ over $\cup$: $A \odot (B \cup C) = A \odot B \cup A \odot C$. Such a representation is of course not unique. Note that below any iterated catenation or iterated shuffle the term C_{ik} might be arbitrarily complex.

Proof. Proceed by structural induction. If L is already finite, or the shuffle or iteration closure of some language in $\mathcal{ER}$, then I is a singleton set and the union contains one trivial product. If $L = M \cup N$, with I_M and I_N wlog disjoint possible index sets of the respective unions, then $L = \bigcup_{i \in I_M \cup I_N} \bigodot_{k=0}^{K(i)} A_{ik}$. If $L = M \odot N$, then $L = \bigcup_{(i,j) \in I_M \times I_N} \bigodot_{k=0}^{K(i)} A_{ik} \odot \bigodot_{k=0}^{K(j)} A_{jk}$, with $|I_M| \times |I_N|$ such products. □

Lemma 8. $(\sqcup, {}^{\sqcup})(\mathcal{FIN}) \not\subseteq \mathcal{ER}$.

Proof. Consider $L = \{abc\}^{\sqcup} \sqcup \{bc\} \notin \mathcal{ER}$. The following property holds:

(1) $\forall w \in L \ : \ ||w||_a + 1 = ||w||_b = ||w||_c$.

Using a representation from lemma 7, L can be written as a finite union indexed by I such that for each $i \in I$, there is a maximum number $m(i)$ of letters contributed by the finite languages:

$$m(i) = \sum_{k \in \{0,\cdots,K(i)\}, |A_{ik}| < \infty} ||A_{ik}||$$

.

Let $m = \max\{m(i) | i \in I\}$.

Furthermore, L has the property that $w \in A_{ik} \Rightarrow |w|_a = |w|_b = |w|_c$ for all infinite sets A_{ik} (otherwise, property (1) will be destroyed by one of the words w_1 obtained by selecting an arbitrary element of each finite set and λ from all infinite sets and w_2 obtained in the same way save for A_{ik}, where we choose a nonbalanced word, as can be seen via the Parikh mapping).

A word $w = a^n b b^n c c^n$ with $n > m$, which is in L for any $n > m$, must be generated by a concatenation of sets A_k such that at least one b is taken from a finite set, say A_{k_b}, likewise one c is taken from a finite set A_{k_c}.

By virtue of the catenation operation's order preserving property, the word w is generated such that

$$w \in (\bigodot_{k<k_b} A_k) \odot A_{k_b} \odot (\bigodot_{k_b<k<k_c} A_k) \odot A_{k_c} \odot (\bigodot_{k_c<k} A_k)$$

.

Obviously, if $n > m$, not all a can come from finite A_k. Hence there must exist some infinite set A_a to the left of A_{k_b} such that $a^\ell \in A_a$ for some $\ell > 0$. A contradiction.

□

$L' = \{ab\}^{\sqcup} \sqcup \{c\} \notin \mathcal{ER}$ is a simpler language with a similar property. It can be shown in a similar way, using $\forall w \in L' \mid ||w||_a = ||w||_b$ and words $a^n c b^n \in L'$.

Lemma 9. $({}^{\sqcup})(\mathcal{FIN}) \not\subseteq (\odot, \cup, \sqcup, {}^{\odot})(\mathcal{FIN}) = \mathcal{REG}$.

Proof. $\{ab\}^{\sqcup}$ is not regular because $\{ab\}^{\sqcup} \cap (\{a\}^{\odot} \odot \{b\}^{\odot})$ is not. □

Lemma 10. $({}^{\odot})(\mathcal{FIN}) \not\subseteq (\odot, \cup, \sqcup, {}^{\sqcup})(\mathcal{FIN})$.

Proof. Consider $L = \{ab\}^{\odot} \in ({}^{\odot})(\mathcal{FIN})$. $L \notin (\cup, \odot, \sqcup, {}^{\sqcup})(\mathcal{FIN})$.

Assume the contrary. Since $||L|| = \infty$ the operation ${}^{\sqcup}$, the only one producing infinite sets, has to be used at least once, $A = B^{\sqcup}$. Let it be the last one in the structural tree representing L. Clearly, $B \neq \emptyset$, $B \neq \{\lambda\}$, and $\exists w \in B : w = uav$. But then $uuaavv \in \{w\} \sqcup \{w\} \subseteq A$. Neither $\odot$ nor $\sqcup$ will erase such a word $u'aav'$. A contradiction. □

3.2 The Upper Hierarchy

In this part we investigate higher important language classes, in particular those defined by systems of equations, and their relations to well known classes. This is illustrated in Figure 2.

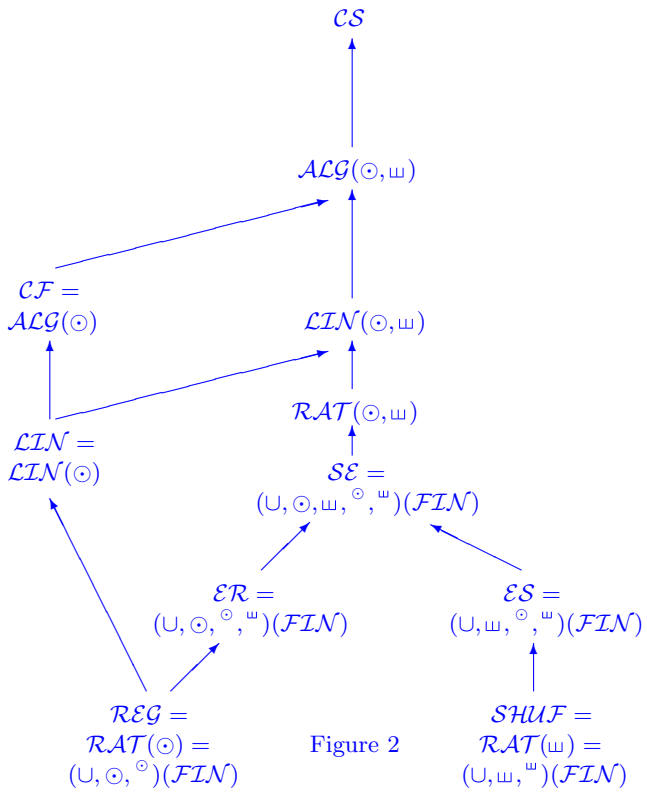

Figure 2

Lemma 11. $\mathcal{SE} \subseteq \mathcal{RAT}(\odot, \sqcup)$

Proof. Construction of a system of equations by structural induction. It suffices to start with singletons.

$\alpha \in \Sigma_1^*$. $X_1 = \alpha$.

$A \cup B$ with Y, Z variables for A, B. Add $X_1 = Y_1, X_1 = Z_1$.

$A \odot B$. Add $X_1 = Z_1$ and $Z = Y_1 \odot \beta$ if $Z = \beta$ is an equation for A. Similar for $A \sqcup B$ with $\odot$ replaced by $\sqcup$.

$A^{\odot}$ with Y variables for A. Add $X_1 = Y_1$ and $Y = X_1 \odot \beta$ if $Y = \beta$ is an equation for A. Similar for $A^{\sqcup}$ with $\odot$ replaced by $\sqcup$. □

Lemma 12. $\mathcal{RAT}(\odot, \sqcup) \not\subseteq \mathcal{SE}$.

Proof. We provide a counterexample.

Example 2. Another example of a rational system of equations is given by
$X = Y \odot \alpha \cup \{\lambda\}\ Y = Z \sqcup \beta\ Z = U \odot \gamma\ U = X \sqcup \delta$
with $\alpha = \{a\}, \beta = \{b\}, \gamma = \{c\}, \delta = \{d\}$.

The least fix point solution X fulfills $X \subseteq \{w \mid ||w||_a = ||w||_b = ||w||_c = ||w||_d\}$ whose words of length $4k, k \in \mathbb{N}$ can be easily calculated: some of them are $d^k b^k (ca)^k$, $(dcba)^k$. Some words not in any solution X are any words ending in d, or words with aa or cc infixes.

We prove that no infinite subset of X containing infinitely many $d^k b^k (ca)^k$ words is in $\mathcal{SE}$.

1) $X = A^{\odot}$ or 2) $X = A^{\sqcup}$ are out, because for 2) no word with an arbitrary number of consecutive c or a is in X; and for 1) eventually some d^i would have to be in A, which can then be added to the end, yielding a word not in X. So any language whose minimal $\mathcal{SE}$ term has depth 1, does not provide the solution X. All other possibilities reduce the question of finding such a $\mathcal{SE}$ language to the question of finding one with a minimal term depth reduced by 1:

$X = A \cup B$ still means that one of A and B still contains infinitely many such words and none contains a word not in X, begging the question.

$X = A \odot B$, with non-trivial A and B, means all $d^k b^k (ca)^k$ words must come from A and B wholesale because otherwise the balance between the numbers of occurrences of a, b, c and d can necessarily be upset by pumping.

$X = A \sqcup B$, with non-trivial A and B, means that if A contains any word ucv, B contains neither $u_2 a v_2$ nor $u_3 c v_3$. Neither can it contain any word with b nor d since no word in X ends with either of these. □

Lemma 13. *(also [4])* $\mathcal{SHUF} \not\subseteq \mathcal{CF}$

Proof. $L = \{abc\}^{\sqcup} \in ({}^{\sqcup})(\mathcal{FIN})$. But since $\mathcal{CF}$ is closed under intersection with regular sets, $L \cap (\{a\}^{\odot} \odot \{b\}^{\odot} \odot \{c\}^{\odot}) = \{a^n b^n c^n \mid n \geq 0\} \notin \mathcal{CF}$. □

To prove the following lemma iteration lemmata for the classes $\mathcal{RAT}(\odot, \sqcup)$, $\mathcal{LIN}(\odot, \sqcup)$ and $\mathcal{ALG}(\odot, \sqcup)$, similar to such for $\mathcal{REG}$, $\mathcal{LIN}$ and $\mathcal{CF}$ are applied. For lack of space they and the following counterxexamples will be presented in another article. For general iteration lemmata see [10].

Lemma 14. $L_1 = \{a^n b^n \mid n \geq 0\} \in \mathcal{LIN}$, $L_1 \notin \mathcal{RAT}(\odot, \sqcup)$,
$L_2 = \{a^m b^m c^n d^n \mid m, n \geq 0\} \in \mathcal{CF}$, $L_2 \notin \mathcal{LIN}(\odot, \sqcup)$,
$L_3 = \{a^n b^n c^n \mid n \geq 0\} \in \mathcal{CS}$, $L_3 \notin \mathcal{ALG}(\odot, \sqcup)$.

Putting together the last lemmata as well as such known for the Chomsky hierarchy and from Figure 1, we get the complete diagram shown in Figure 2.

4 Outlook

In another paper we have investigated structural, closure and decidability properties of language classes presented in this paper, as well as iteration lemmata for them, and their relation to semilinear sets.

It would also be interesting to know for each of the proper inclusions in the diagrams whether it is decidable if a language of the higher family, given by a term of the corresponding type, is also a member of the lower one ([6], p. 104) (e.g. decidability of whether shuffle closure of a regular language is still regular). Also other decidability problems as equivalence should be investigated, as well as the complexity of the language classes considered.

References

1. Câmpeanu, Cezar; Salomaa, Kai; Vágvölgyi, Sándor: *Shuffle Quotient and Decomposition.* Springer **LNCS 2295**, pp. 186–196, 2002.
2. Czaja, Ludwik; Kudlek, Manfred: *Language Theoretic Properties of Client/Server Systems.* Proceedings of CS&P 2011, pp. 79-84, 2011.
3. Ginsburg, Seymour: *The Mathematical Theory of Context-free Languages.* McGraw-Hill, 1966.
4. Gischer, Jay: *Shuffle Languages, Petri Nets, and Context-sensitive Grammars.* **CACM 24 (9)**, pp. 597–605, 1981.
5. Ito, Masami: *Shuffle Decomposition of Regular Languages.* Journal of Universal Computer Science, Vol. 8, No. 2, pp. 257–259, 2002
6. Ito, Masami: *Algebraic Theory of Automata and Languages.* World Scientific, 2004.
7. Jantzen, Matthias: *The Power of Synchronizing Operations on Strings.* Technical Report, FB Informatik, Univ. Hamburg, IfI-HH-B-67/80, 1980.
8. Jantzen, Matthias: *Extending Regular Expressions with Iterated Shuffle.* Technical Report, FB Informatik, Univ. Hamburg, IfI-HH-B-99/84, 1984.
9. Jantzen, Matthias: *Extending Regular Expressions with Iterated Shuffle.* **TCS 38**, pp. 223–247, 1985.
10. Kudlek, Manfred: *On General Iteration Lemmata for Certain Classes of Word, Trace and Graph Languages.* **FI 37 (4)**, pp. 413–422, 1999.
11. Kudlek, Manfred: *On Semilinear Sets over Commutative Semirings.* **FI 79 (3-4)**, pp. 447–452, 2007.
12. Kuich, Werner; Salomaa, Arto: *Semirings, Automata, Languages.* Springer, 1986.