

Algebraic Aspects of Families of Fuzzy Languages

Peter R.J. Asveld

Department of Computer Science, Twente University of Technology
P.O. Box 217, 7500 AE Enschede, the Netherlands
e-mail: `infprja@cs.utwente.nl`

Abstract — We study operations on fuzzy languages such as union, concatenation, Kleene \star , intersection with regular fuzzy languages, and several kinds of (iterated) fuzzy substitution. Then we consider families of fuzzy languages, closed under a fixed collection of these operations, which results in the concept of full Abstract Family of Fuzzy Languages or full AFFL. This algebraic structure is the fuzzy counterpart of the notion of full Abstract Family of Languages that has been encountered frequently in investigating families of crisp (i.e., non-fuzzy) languages. Some simpler and more complicated algebraic structures (such as full substitution-closed AFFL, full super-AFFL, full hyper-AFFL) will be considered as well.

In the second part of the paper we focus our attention to full AFFL's closed under iterated parallel fuzzy substitution, where the iterating process is prescribed by given crisp control languages. Proceeding inductively over the family of these control languages, yields an infinite sequence of full AFFL-structures with increasingly stronger closure properties.

Keywords: fuzzy languages, closure properties, full Abstract Family of Fuzzy Languages (full AFFL), controlled iterated fuzzy substitution, infinite hierarchy.

1 Introduction

When a new family K of formal languages has been introduced, it will be investigated with respect to many properties. Apart from some features that are very specific to K , one usually considers: decidability problems for K , the interrelationship (equality, inclusion, incomparability) of K with other well-known languages families, and the algebraic or closure properties of K . Restricting ourselves to the latter category of properties, we can observe that in the early days of formal language theory (non)closure under each known operation had to be investigated separately. But after a few years researchers realised that some operations are more fundamental than other ones, and that some other operations can be expressed in these fundamental ones: they are “polynomials” over these fundamental operations.

A milestone in this more algebraic approach to families of formal languages has been the introduction in [15] of the notion of full Abstract Family of Languages (full AFL), being a nontrivial family of languages closed under the following operations: union, concatenation, Kleene \star , homomorphism, inverse homomorphism, and intersection with regular languages; cf. [14] for a monograph on this approach. Similar as in ordinary algebra —where one went from groups to semigroups, rings, and fields— full AFL's gave rise to weaker structures (full

trios, full semi-AFL's [14]) and to more powerful ones: full substitution-closed AFL's [16], full super-AFL's [18] and full hyper-AFL's [1].

The aim of the present paper is to investigate to what extent such an approach is fruitful in case we study fuzzy languages rather than ordinary or crisp languages. A fuzzy language is a generalization of a crisp language in the sense that the characteristic function has been replaced by a more general function, the so-called membership function. To be more precise, consider a formal language L over an alphabet Σ : L is completely determined by its characteristic function $\chi_L : \Sigma^* \rightarrow \{0, 1\}$ defined by

$$\chi_L(x) = \begin{cases} 1 & \text{if } x \in L_0, \\ 0 & \text{if } x \notin L_0. \end{cases}$$

A fuzzy language L over Σ is determined by its membership function $\mu_L : \Sigma^* \rightarrow \mathcal{L}$, where \mathcal{L} is a lattice-ordered structure, which is allowed to be somewhat more complex than the simple two-element set $\{0, 1\}$. So a fuzzy language allows for elements that are not completely in or out the language L , i.e., for elements x in Σ^* with $0 < \mu_L(x) < 1$.

Originally, such a fuzzy language L over Σ has been defined in [22] as a fuzzy subset of Σ^* with $\mathcal{L} = [0, 1]$. Subsequently, the real closed interval $[0, 1]$ has been replaced by a more general algebraic structure, e.g., a (completely distributive) complete lattice; cf. [13].

Recently, the interest in fuzzy context-free grammars and their languages revived in an attempt to model grammatical errors and their rôle in robust parsing [3, 4, 8, 9]. Namely, grammatical errors can be modeled by extending a context-free grammar with additional rules that give rise to terminal strings x with $0 < \mu_L(x) < 1$. In other words, $\mu_L(x)$ expresses the degree of perfection of x with respect to a given, extended grammar G [8]. Consequently, the language $L(G)$ may contain "tiny mistakes" (erroneous strings x with $\mu_L(x)$ close to, but unequal to 1) as well as "capital blunders" (strings x with $\mu_L(x)$ close to, but unequal to 0). For such an extended grammar it is possible to design corresponding recognition and parsing algorithms that, apart from their usual job, compute $\mu_L(x)$ from their input x as well [9].

However, in order to treat the accumulation of grammatical errors adequately ("Making an error twice is worse than making it once." "A long sequence of tiny mistakes results in something that looks like a capital blunder."), \mathcal{L} ought to be augmented with an additional operation; so \mathcal{L} became a commutative semigroup provided with a completely distributive complete lattice order [5, 6, 8]; cf. [17, 21].

In the sequel we restrict ourselves to algebraic or closure properties of families of fuzzy languages. So after some preliminaries on fuzzy sets and on the codomain \mathcal{L} of membership functions (Section 2), we define in Section 3 fuzzy languages and some operations on fuzzy languages. Section 4 is devoted to some families of fuzzy languages such as the family FIN_f of finite fuzzy languages and the family REG_f of regular fuzzy languages. Particularly, this latter family receives a considerable amount of attention in Section 4 because of its predominant part in the study of (families of) fuzzy languages [30, 31]; cf. the position of the family REG of (ordinary) regular languages in the theory of crisp formal languages.

In passing we mention that REG_f is *not* our starting point, as REG is in the theory of full AFL's [14]. Instead we start with the family FIN_f of finite fuzzy languages and we generalize it to an algebraic structure called *fuzzy prequasoid* (Section 5). The reason for taking FIN_f rather than REG_f as initial point is the following. Most operations in Sections 6 and 7 are derived from grammatical devices such as ETOL-system, context-free grammars, and non-self-embedding context-free grammars. Hence, it is more natural to consider, for

instance, fuzzy context-free grammars (formulated in terms of FIN_f) as starting point rather than fuzzy grammars in Backus Normal Form (which are in essence based on REG_f). In this context it is also useful to mention that as soon as a fuzzy prequasoid contains an infinite fuzzy language, it includes REG_f (Lemma 5.2).

Then we consider fuzzy prequasoids closed under regular fuzzy substitution and under substitution into REG_f . This yields a structure that is equivalent to the fuzzy analogue of full AFL: the full Abstract Family of Fuzzy Languages of full AFFL (Section 5), i.e., a nontrivial family of fuzzy languages closed under union, concatenation, Kleene \star , fuzzy homomorphisms, inverse fuzzy homomorphisms, and intersection with regular fuzzy languages.

In Section 6 we give an overview of algebraic structures that are stronger than full AFFL's: full substitution-closed AFFL's [6], full super-AFFL's (full AFFL's closed under nested iterated fuzzy substitution) [8] and full hyper-AFFL's (full AFFL's closed under iterated fuzzy substitution) [5]. The reason that we recall these results in Section 6 is twofold. First, these results heavily rely on Sections 3–5 of the present paper or, phrased otherwise, they are applications of the approach in Sections 3–5. (In this respect the present paper and [5, 6, 8] are companions.) Secondly, in Section 7 we derive results very similar to those in Section 6 and we want to make this correspondence as clear as possible.

Finally, we define in Section 7 an infinite sequence of algebraic structures, each of which is “stronger” than its predecessor in this sequence, while all elements in the sequence are full hyper-AFFL's. This sequence is obtained by (i) controlling the iteration of fuzzy substitutions by crisp control languages that prescribe the order of applying the fuzzy substitutions, and (ii) proceeding inductively over the families of crisp control languages. The last section (Section 8) consists of a few concluding remarks.

2 Preliminaries

We assume familiarity with basic definitions and results from formal language theory; cf. [19, 20, 26, 29] for basic texts and [14] for operations on languages. We also use the rudiments of lattice theory which can be found in many books on algebra; a summary of the relevant concepts is also included in [2].

A *fuzzy set* S , or rather a fuzzy subset S of some universal set U , is given by a function $\mu_S : U \rightarrow \mathcal{L}$; the function μ_S is called the *membership function* of S . The codomain \mathcal{L} of such a membership function is a complete lattice $(\mathcal{L}, \wedge, \vee, 0, 1)$, sometimes provided with additional restrictions; cf. Definition 2.1 below. However, in many papers dealing with fuzzy sets, \mathcal{L} is restricted to the special case of the real closed interval $[0, 1]$. To reduce the number of subscript levels, we often write $\mu(x; S)$ rather than $\mu_S(x)$ in the sequel.

Note that in dealing with fuzzy sets, the membership function $\mu_S : U \rightarrow \mathcal{L}$ is the principal entity, whereas S —in the sense of the support of μ_S , i.e., that part of U where μ_S does not vanish— is actually a derived concept. In the literature S frequently denotes the fuzzy set S as well as the support of μ_S . We will avoid this ambiguity by using a special notation for the latter case. So for each fuzzy set S , its *support* $s(S)$, or rather the support of μ_S , is defined by $s(S) = \{x \in U \mid \mu(x; S) > 0\}$ where $0 = \bigwedge \mathcal{L}$. Another, important derived notion is the *crisp part* $c(S)$ of S , defined by $c(S) = \{x \in U \mid \mu(x; S) = 1\}$ where $1 = \bigvee \mathcal{L}$. An ordinary, non-fuzzy set coincides with its crisp part, i.e., for such a set S , we have $s(S) = c(S)$. Sets satisfying this condition are also called *crisp sets*; their membership function may be viewed as their characteristic function. Notice that for each fuzzy set S , both $s(S)$ and $c(S)$ are crisp sets.

The union $S \cup T$ and the intersection $S \cap T$ of fuzzy sets S and T are defined as usual, i.e., $\mu(x; S \cup T) = \mu(x; S) \vee \mu(x; T)$, and $\mu(x; S \cap T) = \mu(x; S) \wedge \mu(x; T)$, for all x in U . The support of the union equals the union of the supports: $s(S \cup T) = s(S) \cup s(T)$. Although we have $s(S \cap T) \subseteq s(S) \cap s(T)$, equality does not hold in general; cf. Example 3.3. Similarly, for crisp parts we have $c(S \cap T) = c(S) \cap c(T)$ and $c(S \cup T) \supseteq c(S) \cup c(T)$. In the latter case equality does not hold in general (Example 3.3). However, if \mathcal{L} is linearly ordered, then we have $s(S \cap T) = s(S) \cap s(T)$, and $c(S \cup T) = c(S) \cup c(T)$.

Equality of fuzzy sets can be defined in several ways. Henceforth, we use full equality: two fuzzy sets S and T (both being fuzzy subsets of U) are *fully equal*, denoted by $S \doteq T$, if $\mu_S = \mu_T$, i.e., if for all $x \in U$, $\mu(x; S) = \mu(x; T)$. Of course, full equality ($S \doteq T$) implies equality of supports ($s(S) = s(T)$) and of crisp parts ($c(S) = c(T)$), but not vice versa.

Apart from equality, there is also inclusion of fuzzy sets. We will consider two different notions of inclusion for fuzzy sets, called soft and sharp inclusion.

So let S and T be fuzzy subsets of some universal set U . The usual inclusion relation, which we will call the *soft inclusion* of S in T , denoted by $S \subseteq T$, is defined by

$$S \subseteq T \iff \forall x \in U : \mu(x; S) \leq \mu(x; T).$$

Defining the power set $\mathbb{F}(T)$ of T by $\mathbb{F}(T) = \{S \mid S \subseteq T\}$ implies $2^{\#T} \leq \#\mathbb{F}(T) \leq (\#\mathcal{L})^{\#T}$. Here $\#X$ is the cardinality of the fuzzy set X , i.e., the cardinality of the support $s(X)$ of X .

On the other hand, we define the *sharp inclusion* of S in T , written as $S \sqsubseteq T$, by

$$S \sqsubseteq T \iff \forall x \in U : \mu(x; S) > 0 \Rightarrow \mu(x; S) = \mu(x; T).$$

When we define the corresponding concept of power set $\mathbb{P}(T)$ of T by $\mathbb{P}(T) = \{S \mid S \sqsubseteq T\}$, we obtain $\#\mathbb{P}(T) = 2^{\#T}$. Consequently, the power set $\mathbb{P}(T)$ of a finite fuzzy set T is a crisp, finite collection of finite fuzzy sets.

Clearly, the soft inclusions $S \subseteq T$ and $T \subseteq S$ imply $S \doteq T$, as the sharp inclusions $S \sqsubseteq T$ and $T \sqsubseteq S$ do. And, of course, $S \sqsubseteq T$ implies $S \subseteq T$.

In this paper we study a special type of fuzzy sets, viz. fuzzy languages; so S is a language L over some alphabet Σ and U equals the set Σ^* consisting of all strings over Σ . Set-theoretical operations, like union and intersection, will be used in this context; see Section 3.

The remaining part of this section is devoted to the structure of \mathcal{L} . Instead of the closed real interval $[0, 1]$ as in [22], we take a more general structure as codomain of membership functions for fuzzy languages [5, 6, 8, 9]. This structure has been inspired by similar ones in [17, 30, 31, 21].

Definition 2.1. An algebraic structure \mathcal{L} or $(\mathcal{L}, \wedge, \vee, 0, 1, \star)$ is a *type-00 lattice* if

- $(\mathcal{L}, \wedge, \vee, 0, 1)$ is a completely distributive complete lattice. So $a \wedge \bigvee_i b_i = \bigvee_i (a \wedge b_i)$ and $(\bigvee_i a_i) \wedge b = \bigvee_i (a_i \wedge b)$ hold for all a_i, a, b_i and b in \mathcal{L} . And 0 and 1 are the smallest and the greatest element of \mathcal{L} , respectively: $0 = \bigwedge \mathcal{L}$ and $1 = \bigvee \mathcal{L}$.
- (\mathcal{L}, \star) is a commutative semigroup.
- The following identities hold for all a and b in \mathcal{L} :

$$\begin{aligned} a \star \bigvee_i b_i &= \bigvee_i (a \star b_i), \\ (\bigvee_i a_i) \star b &= \bigvee_i (a_i \star b), \\ 0 \wedge a &= 0 \star a = a \star 0 = 0, \\ 1 \wedge a &= 1 \star a = a \star 1 = a. \end{aligned}$$

A type-00 lattice in which the operation \star coincides with \wedge is called a *type-01 lattice*: so it is a completely distributive complete lattice. A *type-10 lattice* is a type-00 lattice in which $(\mathcal{L}, \wedge, \vee, 0, 1)$ is a totally ordered set or chain, i.e., for all a and b in \mathcal{L} , we have $a \wedge b = a$ or $a \wedge b = b$. In a type-10 lattice the operations \vee and \wedge are usually denoted by \max and \min , respectively. Finally, when \mathcal{L} is both a type-01 lattice and a type-10 lattice, \mathcal{L} is called a *type-11 lattice*. \square

Lemma 2.2. [5, 6] *In each type-00 lattice \mathcal{L} , we have for all $a, b \in \mathcal{L}$, $a \star b \leq a \wedge b$.*

Proof. By the distributivity of \star over \vee , $a \star (1 \vee b) = a \star 1 \vee a \star b$ holds. As $1 \vee b = 1$ and $a \star 1 = a$, this yields $a = a \vee a \star b$; so $a \star b \leq a$. Similarly, $a \star b \leq b$, and hence $a \star b \leq a \wedge b$. \square

Corollary 2.3. *Let \mathcal{L} be a type-00 lattice. If $a \star b = 1$, then $a = b = 1$.*

Proof. By Lemma 2.2, we have $1 = a \star b \leq a \wedge b \leq a \leq \bigvee \mathcal{L} = 1$, and similarly for b . \square

Example 2.4. Let $[0, 1]$ be the closed interval of real numbers in between 0 and 1.

(1) Then $([0, 1] \times [0, 1], \wedge, \vee, (0, 0), (1, 1), \star)$ with $(x_1, y_1) \wedge (x_2, y_2) = (\min\{x_1, x_2\}, \min\{y_1, y_2\})$, $(x_1, y_1) \vee (x_2, y_2) = (\max\{x_1, x_2\}, \max\{y_1, y_2\})$ and $(x_1, y_1) \star (x_2, y_2) = (x_1 x_2, y_1 y_2)$ for all x_1, x_2, y_1 and y_2 in $[0, 1]$ is a type-00 lattice.

(2) Let \mathcal{L} be $(\{0, \xi, \eta, 1\}, \wedge, \vee, 0, 1, \wedge)$ with $0 < \xi < 1$, $0 < \eta < 1$, and ξ and η are incomparable. Then \mathcal{L} is a type-01 lattice (and it is the 4-element distributive lattice that is not a chain).

(3) $([0, 1], \min, \max, 0, 1, \star)$ with $x_1 \star x_2 = x_1 x_2$ for all x_1 and x_2 in $[0, 1]$ is a type-10 lattice.

(4) $([0, 1], \min, \max, 0, 1, \min)$ is a type-11 lattice. \square

In practical examples the real closed interval $[0, 1]$ is usually restricted to (i.e., replaced by) the set of its computable or even its rational elements; cf. [9]. We refer to [13] for the impact of computability constraints in fuzzy formal languages.

3 Fuzzy Languages and Operations on Fuzzy Languages

A fuzzy language L is a fuzzy subset of Σ^* where Σ is the alphabet of L . So Σ^* plays the rôle of universal set for L .

Definition 3.1. Let \mathcal{L} be a type-00 lattice and let Σ be an alphabet. A \mathcal{L} -fuzzy language L over Σ is a \mathcal{L} -fuzzy subset of Σ^* , i.e., it is a pair (Σ, μ_L) where μ_L is a function $\mu_L : \Sigma^* \rightarrow \mathcal{L}$, the *membership function* of L . For each \mathcal{L} -fuzzy language L , $s(L)$ and $c(L)$ denote the *support* and the *crisp part* of L , respectively: $s(L) = \{w \in \Sigma^* \mid \mu(w; L) > 0\}$ and $c(L) = \{w \in \Sigma^* \mid \mu(w; L) = 1\}$.

When \mathcal{L} is clear from the context, we use “fuzzy language” instead of “ \mathcal{L} -fuzzy language”.

Each ordinary (non-fuzzy) language L satisfies $s(L) = c(L)$. So an ordinary language is also called a *crisp language*. \square

In defining a fuzzy language L over Σ , as in the following example, we always restrict ourselves to specifying the values of μ_L for elements of $s(L)$ only. So if $\mu(x; L)$ is not given for a particular x , it is tacitly assumed that $x \in \Sigma^* - s(L)$, and hence we have $\mu(x; L) = 0$.

Example 3.2. (1) Let \mathcal{L} be the type-00 lattice of Example 2.4(1), and consider the \mathcal{L} -fuzzy language L_0 over $\{a, b\}$, defined by

$$\mu(a^m b^n a^m; L_0) = \left(\frac{m}{\max\{1, m, n\}}, \frac{n}{\max\{1, m, n\}} \right) \quad \text{if } m, n \geq 1.$$

Then the crisp part of L_0 equals $c(L_0) = \{a^m b^n a^m \mid m \geq 1\}$: for each x in $c(L_0)$, we have $\mu(x; L_0) = (1, 1)$.

(2) Consider the type-01 lattice \mathcal{L} of Example 2.4(2) and the \mathcal{L} -fuzzy languages L_1 and L_2 over $\{a, b\}$ defined by $\mu(a^m b^m a^n; L_1) = \xi$ and $\mu(a^m b^n a^n; L_2) = \eta$ for $m, n \geq 1$. Then $c(L_1) = c(L_2) = \emptyset$ but both L_1 and L_2 are nonempty languages since $s(L_1) = \{a^m b^m a^n \mid m, n \geq 1\}$ and $s(L_2) = \{a^m b^n a^n \mid m, n \geq 1\}$.

(3) Let again \mathcal{L} be the type-01 lattice of Example 2.4(2). As a slight variation of the previous example, define the \mathcal{L} -fuzzy languages L_3 and L_4 over $\{a, b, c, d\}$ by $\mu(a^n b^n c^m d^m; L_3) = \xi$ and $\mu(a^n b^m c^m d^n; L_4) = \eta$ for $m, n \geq 1$. Of course, we have $c(L_3) = c(L_4) = \emptyset$, and both L_3 and L_4 are nonempty languages. \square

Next we turn to some operations on fuzzy languages. First, we recall the operations union, intersection, concatenation, Kleene $+$ and Kleene \star for \mathcal{L} -fuzzy languages defined in [5, 6, 8]. We use λ to denote the empty word.

Let $L_1 = (\Sigma_1, \mu_{L_1})$ and $L_2 = (\Sigma_2, \mu_{L_2})$ be fuzzy languages, then the *union*, the *intersection*, and the *concatenation* of the fuzzy languages L_1 and L_2 , denoted by $L_1 \cup L_2 = (\Sigma_1 \cup \Sigma_2, \mu_{L_1 \cup L_2})$, $L_1 \cap L_2 = (\Sigma_1 \cap \Sigma_2, \mu_{L_1 \cap L_2})$ and $L_1 L_2 = (\Sigma_1 \cup \Sigma_2, \mu_{L_1 L_2})$ respectively, are defined by

$$\begin{aligned} \mu(x; L_1 \cup L_2) &= \mu(x; L_1) \vee \mu(x; L_2), \\ \mu(x; L_1 \cap L_2) &= \mu(x; L_1) \wedge \mu(x; L_2), \text{ and} \\ \mu(x; L_1 L_2) &= \bigvee \{ \mu(y; L_1) \star \mu(z; L_2) \mid x = yz \}, \end{aligned}$$

for all x in $(\Sigma_1 \cup \Sigma_2)^*$.

Example 3.3. (1) For the union and the intersection of the fuzzy languages L_1 and L_2 of Example 3.2(2), we have

$$\mu(x; L_1 \cup L_2) = \begin{cases} 1 & \text{if } x = a^m b^m a^m \text{ for some } m \geq 1, \\ \xi & \text{if } x = a^m b^m a^n \text{ and } m \neq n \text{ } (m, n \geq 1), \\ \eta & \text{if } x = a^m b^n a^n \text{ and } m \neq n \text{ } (m, n \geq 1), \end{cases}$$

and $L_1 \cap L_2 = s(L_1 \cap L_2) = c(L_1 \cap L_2) = \emptyset$, respectively. Note that $s(L_1) \cap s(L_2) \neq \emptyset$, and $c(L_1 \cup L_2) = c(L_0) \neq \emptyset$ (L_0 is the fuzzy language from Example 3.2(1)), whereas $c(L_1) = c(L_2) = \emptyset$.

(2) Similarly, for the union of L_3 and L_4 of Example 3.2(3), we get

$$\mu(x; L_3 \cup L_4) = \begin{cases} 1 & \text{if } x = a^n b^n c^n d^n \text{ for some } n \geq 1, \\ \xi & \text{if } x = a^n b^n c^m d^m \text{ and } m \neq n \text{ } (m, n \geq 1), \\ \eta & \text{if } x = a^n b^m c^m d^n \text{ and } m \neq n \text{ } (m, n \geq 1). \end{cases}$$

We return to these unions in Example 6.4(2) and in Section 8 below. \square

The operations of Kleene $+$ and Kleene \star for a fuzzy language $L = (\Sigma, \mu_L)$ are defined by

$$\begin{aligned} L^+ &\doteq L \cup LL \cup LLL \cup \dots \doteq \bigcup \{ L^i \mid i \geq 1 \} \quad \text{and} \\ L^* &\doteq \{ \lambda \} \cup L \cup LL \cup \dots \doteq \bigcup \{ L^i \mid i \geq 0 \}, \end{aligned}$$

respectively, where $L^0 \doteq \{ \lambda \}$, and $L^{n+1} \doteq L^n L$ with $n \geq 0$ [5, 6, 8]. Then, for $n \geq 0$ we have

$$\begin{aligned} \mu(x; L^n) &= \bigvee \{ \mu(x_1; L) \star \mu(x_2; L) \star \dots \star \mu(x_n; L) \mid x_1 x_2 \dots x_n = x \}, \quad \text{and} \\ \mu(x; L^*) &= \bigvee \{ \mu(x_1; L) \star \mu(x_2; L) \star \dots \star \mu(x_n; L) \mid n \geq 0, x_1 x_2 \dots x_n = x \}. \end{aligned}$$

Thus $\mu(\lambda; L^0) = 1$, as $x_1 x_2 \dots x_n = \lambda$ and $a_1 \star a_2 \star \dots \star a_n = 1$ ($a_1, \dots, a_n \in \mathcal{L}$) in case $n = 0$. Consequently, $\mu(\lambda; L^*) = 1$, and $L^* \doteq L^+ \cup \{ \lambda \}$ where the latter set in this union is a crisp set.

Remark. In order to avoid all kinds of technical difficulties in the sequel, we will make the following convention. If a fuzzy language L contains the empty word λ , then $\mu(\lambda; L) = 1$. Hence for an arbitrary fuzzy language L , we have either $\mu(\lambda; L) = 1$ or $\mu(\lambda; L) = 0$. \square

Other operations on fuzzy languages, like homomorphisms and substitutions, are defined as fuzzy functions on fuzzy languages. A fuzzy function is a special instance of a fuzzy relation. A *fuzzy relation* R between crisp sets X and Y is a fuzzy subset of $X \times Y$. If $R \subseteq X \times Y$ and $S \subseteq Y \times Z$ are fuzzy relations, then their composition $R \circ S$ is defined by

$$\mu((x, z); R \circ S) = \bigvee \{ \mu((x, y); R) \star \mu((y, z); S) \mid y \in Y \}. \quad (1)$$

Then a *fuzzy function* $f : X \rightarrow Y$ is a fuzzy relation $f \subseteq X \times Y$, satisfying the restriction that for all x in X : if $\mu((x, y); f) > 0$ and $\mu((x, z); f) > 0$ hold, then $y = z$ and hence $\mu((x, y); f) = \mu((x, z); f)$. For fuzzy functions (1) holds as well, but we write the composition of two functions $f : X \rightarrow Y$ and $g : Y \rightarrow Z$ as $g \circ f : X \rightarrow Z$ rather than as $f \circ g$.

Remember that $\mathbb{F}(X)$ denotes the power set of the fuzzy set X , i.e., the collection of all fuzzy subsets of the fuzzy set X . In the sequel we will encounter functions $f : V^* \rightarrow \mathbb{F}(V^*)$ that will be extended to $f : \mathbb{F}(V^*) \rightarrow \mathbb{F}(V^*)$ by $f(L) = \bigcup \{ f(x) \mid x \in L \}$ and for each fuzzy subset L of V^* ,

$$\mu(y; f(L)) = \bigvee \{ \mu(x; L) \star \mu((x, y); f) \mid x \in V^* \}. \quad (2)$$

Consequently, by (1) and (2) fuzzy functions like $f \circ f$, $f \circ f \circ f$, and so on, which are obtained by iterating the function f , are now defined. Clearly, each of these functions $f^{(k)}$ is of type $f^{(k)} : \mathbb{F}(V^*) \rightarrow \mathbb{F}(V^*)$. A finite set $\{f_1, \dots, f_n\}$ of such functions can be iterated in the same way; cf. Definitions 6.1, 6.2, 6.3 and 7.1 below.

4 Families of Fuzzy Languages

Let Σ_ω denote a countably infinite set of symbols. All families of languages that we will consider in the sequel, only use symbols from this set. And \mathcal{L} is a type-00 lattice except when stated otherwise.

Definition 4.1. A *family of fuzzy languages* K is a set of fuzzy languages $L = (\Sigma_L, \mu_L)$ such that each Σ_L is a finite subset of Σ_ω . We assume that for each fuzzy language (Σ_L, μ_L) in K , the alphabet Σ_L is minimal, i.e., a symbol α belongs to Σ_L if and only if there exists a word w in which α occurs and for which $\mu(w; L) > 0$ or, equivalently, for which $w \in s(L)$.

A family K of fuzzy languages is called *nontrivial* if K contains a nontrivial language, i.e., a language (Σ_L, μ_L) with $s(L) \cap \Sigma_L^+ \neq \emptyset$ or, equivalently, $\mu(x; L) > 0$ for some $x \in \Sigma_L^+$.

And K is called *normalized* if it contains a normalized language, i.e., a language (Σ_L, μ_L) with $c(L) \cap \Sigma_L^+ \neq \emptyset$ or, equivalently, $\mu(x; L) = 1$ for some $x \in \Sigma_L^+$.

For each family K of fuzzy languages, the *crisp part* $c(K)$ of K is the family of crisp languages defined by $c(K) = \{c(L) \mid L \in K\}$. \square

Example 4.2. The fuzzy languages L_i ($0 \leq i \leq 4$) from Example 3.2 are all nontrivial; L_0 is normalized, but L_1 , L_2 , L_3 and L_4 are not. Note that both $L_1 \cup L_2$ and $L_3 \cup L_4$ are normalized too (Example 3.3). \square

Henceforth, we assume that each family K of fuzzy languages is normalized and closed under isomorphism (“renaming of symbols”), i.e., for each language L in K over some alphabet Σ_L and for each bijective non-fuzzy mapping $i : \Sigma_L \rightarrow \Sigma'_L$ —extended to words and to

languages in the usual way— we have that the language $i(L)$ also belongs to K . Remark that for all x in Σ_L^* , we have $\mu(x; L) = \mu(i(x); i(L))$ or, equivalently, $L \overset{\circ}{=} i(L)$.

Among the most simple normalized families of fuzzy languages, we have the family FIN_f of finite fuzzy languages

$$\text{FIN}_f = \{L \mid s(L) = \{w_1, w_2, \dots, w_n\}, w_i \in \Sigma_\omega^*, 1 \leq i \leq n; n \geq 0\},$$

the family ONE_f of singleton fuzzy languages

$$\text{ONE}_f = \{L \mid s(L) = \{w\}, w \in \Sigma_\omega^*\},$$

the family ALPHA_f of fuzzy alphabets

$$\text{ALPHA}_f = \{L \mid s(L) = \Sigma, \Sigma \subset \Sigma_\omega, \Sigma \text{ is finite}\},$$

and the family SYMBOL_f of singleton fuzzy alphabets

$$\text{SYMBOL}_f = \{L \mid s(L) = \{\alpha\}, \alpha \in \Sigma_\omega\}.$$

The crisp counterparts of these language families are $\text{FIN} = \{\{w_1, w_2, \dots, w_n\} \mid w_i \in \Sigma_\omega^*, 1 \leq i \leq n; n \geq 0\}$, $\text{ONE} = \{\{w\} \mid w \in \Sigma_\omega^*\}$, $\text{ALPHA} = \{\Sigma \mid \Sigma \subset \Sigma_\omega, \Sigma \text{ is finite}\}$, and $\text{SYMBOL} = \{\{\alpha\} \mid \alpha \in \Sigma_\omega\}$, respectively; cf. Lemma 4.4 below.

The family of regular fuzzy languages is denoted by REG_f ; it is defined in a way very similar to its crisp counterpart REG .

Definition 4.3. The family of regular fuzzy languages REG_f is the smallest set satisfying:

- The fuzzy subsets \emptyset and $\{\lambda\}$ of \emptyset^* belong to REG_f . Note that $\mu(\lambda; \{\lambda\}) = \mu(\lambda; \emptyset^*) = 1$.
- For each σ in Σ_ω , the fuzzy subset $\{\sigma\}$ of $\{\sigma\}^*$ belongs to REG_f .
- If R_1 and R_2 are in REG_f , then so are $R_1 \cup R_2$, $R_1 R_2$, and R_1^* . □

Lemma 4.4. (1) $c(\text{FIN}_f) = \text{FIN}$, $c(\text{ONE}_f) = \text{ONE} \cup \{\emptyset\}$, $c(\text{ALPHA}_f) = \text{ALPHA}$, and $c(\text{SYMBOL}_f) = \text{SYMBOL} \cup \{\emptyset\}$.

(2) If \mathcal{L} is a type-10 lattice, then for \mathcal{L} -fuzzy languages, $c(\text{REG}_f) = \text{REG}$.

Proof. The equalities under (1) are straightforward, and the inclusion $\text{REG} \subseteq c(\text{REG}_f)$ is obvious. The converse inclusion $c(\text{REG}_f) \subseteq \text{REG}$ can easily be established by induction over the structure of a regular fuzzy language (Definition 4.3) using Corollary 2.3 and the fact that $c(R_1 \cup R_2) = c(R_1) \cup c(R_2)$ holds for linearly ordered \mathcal{L} . □

Example 4.5. The equality $c(R_1 \cup R_2) = c(R_1) \cup c(R_2)$ does not hold in general for arbitrary type-00 lattices. Our argument is based on (i) the structure of the simplest type-00 lattice that is not linearly ordered (Example 2.4(2)), and (ii) the ambiguous description of certain regular languages; cf. [10].

So consider the regular fuzzy languages L_5 and L_6 over $\{a\}$ defined by $\mu(a^{2n}; L_5) = \xi$, $\mu(a^{3n}; L_6) = \eta$ ($n \geq 1$), and $\mu(\lambda; L_5) = \mu(\lambda; L_6) = 1$. Then $c(L_5) = c(L_6) = \{\lambda\}$ and $c(L_5 \cup L_6) = \{a^{6n} \mid n \geq 0\}$. Hence $c(L_5) \cup c(L_6) \subset c(L_5 \cup L_6)$. □

Closely related to regular fuzzy languages is a kind of fuzzy finite automaton. Many variations of the finite-state concept for fuzzy languages have been introduced of which we mention but a few: [23, 24, 27, 28, 30, 31]. The next definition and equivalence result (Proposition 4.9) is useful but not surprising.

Definition 4.6. A nondeterministic fuzzy finite automaton with λ -moves or NFFA M is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$, where Q is a crisp finite set of states, Σ is an alphabet, q_0 is an element of Q , F is a crisp subset of the crisp set Q , and δ is a fuzzy function of type $\delta : Q \times (\Sigma \cup \{\lambda\}) \rightarrow \mathbb{F}(Q)$ that satisfies the following condition: for each q in Q , $\delta(q, \lambda)$ is a crisp subset of Q .

The function δ is extended to $\hat{\delta} : Q \times \Sigma^* \rightarrow \mathbb{F}(Q)$ as follows: for all q in Q , $\hat{\delta}(q, \lambda) \doteq \delta(q, \lambda)$ and

$$\hat{\delta}(q, \sigma\omega) \doteq \bigcup \{ \hat{\delta}(q', \omega) \mid q' \in \delta(q, \sigma) \},$$

that is, according to (2),

$$\mu(p; \hat{\delta}(q, \sigma\omega)) = \bigvee \{ \mu(p; \hat{\delta}(q', \omega)) \star \mu(q'; \delta(q, \sigma)) \mid q' \in Q \} \quad (p \in Q).$$

The fuzzy language $L(M)$ accepted by the NFFA M is defined by

$$L(M) \doteq \{ x \in \Sigma^* \mid \hat{\delta}(q_0, x) \cap F \neq \emptyset \}$$

or, equivalently,

$$\mu(x; L(M)) = \bigvee \{ \mu(q; \hat{\delta}(q_0, x)) \mid q \in F \}.$$

Two NFFA's M_1 and M_2 are called *equivalent* if $L(M_1) \doteq L(M_2)$. \square

Henceforth we use expressions like $X = \{ \dots, x/\mu(x; X), \dots \}$ to denote finite fuzzy sets (including the degrees of membership) concisely.

Example 4.7. Let \mathcal{L} be the type-01 lattice defined in Example 2.4(2). Consider the NFFA $M = (Q, \Sigma, \delta, q_0, F)$ with $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$, $\Sigma = \{a\}$, $F = \{q_0, q_1, q_3\}$ and δ is defined by

$$\begin{aligned} \delta(q_0, \lambda) &= \{q_1/1, q_3/1\}, & \delta(q_1, a) &= \{q_2/1\}, & \delta(q_2, a) &= \{q_1/\xi\}, \\ \delta(q_3, a) &= \{q_4/1\}, & \delta(q_4, a) &= \{q_5/1\}, & \delta(q_5, a) &= \{q_3/\eta\}. \end{aligned}$$

Then the \mathcal{L} -fuzzy language $L(M)$ satisfies $L(M) \doteq L_5 \cup L_6$ where L_5 and L_6 are the fuzzy regular languages defined in Example 4.5. \square

Lemma 4.8. *Let $M = (Q, \Sigma, \delta, q_0, F)$ be an NFFA. Then there is an equivalent NFFA $M' = (Q', \Sigma, \delta', q'_0, \{f\})$ such that $Q' = Q \cup \{q_0, f\}$, the in-degree of q_0 is zero and the out-degree of f is zero, i.e., δ' is a fuzzy function of type $\delta' : (Q \cup \{q'_0, f\}) \times (\Sigma \cup \{\lambda\}) \rightarrow \mathbb{F}(Q \cup \{f\})$ with $\delta'(f, \lambda) \doteq \{f/1\}$ and $\forall \alpha \in \Sigma : \delta'(f, \alpha) \doteq \emptyset$.*

Proof. In order to obtain δ' we extend the fuzzy function δ , viewed as fuzzy relation, by $\delta' \doteq \delta \cup \{((q'_0, \lambda), q_0)/1\} \cup \{((q, \lambda), f)/1 \mid q \in F\}$. \square

Proposition 4.9. *A fuzzy language L is regular if and only if L is accepted by a nondeterministic fuzzy finite automaton.*

Proof. Suppose R is a regular fuzzy language. If R equals \emptyset , $\{\lambda/1\}$ or $\{\sigma/\xi\}$ (cf. Definition 4.3), we define $M = (Q, \Sigma, \delta, q_0, F)$ respectively by

$$\begin{aligned} \emptyset: & \quad Q = \{q_0, q_1\}, \quad F = \{q_1\}, \quad \delta \doteq \emptyset, \\ \{\lambda/1\}: & \quad Q = \{q_0, q_1\}, \quad F = \{q_1\}, \quad \delta(q_0, \lambda) \doteq \{q_1/1\}, \\ \{\sigma/\xi\}: & \quad Q = \{q_0, q_1\}, \quad F = \{q_1\}, \quad \delta(q_0, \sigma) \doteq \{q_1/\xi\}. \end{aligned}$$

Then for each of these three cases we have $R \doteq L(M)$.

Next, let R be equal to $R_1 \cup R_2$, $R_1 R_2$ or R_1^* (Definition 4.3). Suppose $R_i \doteq L(M_i)$ for $i = 1, 2$ with $M_i = (Q_i, \Sigma_i, \delta_i, q_{i0}, \{f_i\})$ satisfying the properties of Lemma 4.8, and $Q_1 \cap Q_2 = \emptyset$. We construct $M_j = (Q_1 \cup Q_2 \cup \{q_{j0}\}, \Sigma_j, \delta_j, q_{j0}, F_j)$ for $j = 3, 4, 5$ by

$$\begin{aligned} R_1 \cup R_2: & \quad F_3 = \{f_1, f_2\}, \quad \delta_3 \doteq \delta_1 \cup \delta_2 \cup \{((q_{30}, \lambda), q_{10})/1, ((q_{30}, \lambda), q_{20})/1\}, \\ R_1 R_2: & \quad F_4 = \{f_2\}, \quad \delta_4 \doteq \delta_1 \cup \delta_2 \cup \{((q_{40}, \lambda), q_{10})/1, ((f_1, \lambda), q_{20})/1\}, \\ R_1^*: & \quad F_5 = \{f_1\}, \quad \delta_5 \doteq \delta_1 \cup \{((q_{50}, \lambda), f_1)/1, ((q_{50}, \lambda), q_{10})/1, ((f_1, \lambda), q_{10})/1\}. \end{aligned}$$

Then $L(M_3) \doteq R_1 \cup R_2$, $L(M_4) \doteq R_1 R_2$, and $L(M_5) \doteq R_1^*$.

The converse implication can easily be established by adapting the standard construction (cf. e.g., [29] pp. 200–203). From Section 3 it will be clear how to apply the operations \vee and \star in updating the degree of membership when we meet a union, a concatenation or a Kleene \star operation in that construction. \square

Other families of fuzzy languages are obtained by applying the operation of fuzzy substitution or some of its generalizations (Definitions 4.10, 5.7, 6.6 and 7.5 below). Fuzzy substitution plays the principal rôle in our approach; it is a straightforward extension of the notion of substitution for crisp languages.

Definition 4.10. Let K be a family of fuzzy languages and let V be an alphabet. A *fuzzy K -substitution* on V is a mapping $\tau : V \rightarrow K$; it is extended to words over V by $\tau(\lambda) \doteq \{\lambda/1\}$ and $\tau(\alpha_1 \cdots \alpha_n) \doteq \tau(\alpha_1) \cdots \tau(\alpha_n)$ with $\alpha_i \in V$ ($1 \leq i \leq n$), and to languages by $\tau(L) \doteq \bigcup \{\tau(w) \mid w \in L\}$.

If for each $\alpha \in V$, $\tau(\alpha) \subseteq V^*$, then $\tau : V \rightarrow K$ is called a *fuzzy K -substitution over V* .

If we have $\alpha \in \tau(\alpha)$ for each $\alpha \in V$, then $\tau : V \rightarrow K$ is called a *nested fuzzy K -substitution*.

If the family K equals FIN_f or REG_f , τ is called a *fuzzy finite* or a *fuzzy regular substitution*, respectively.

Given families K and K' of fuzzy languages, let $\text{Sùb}(K, K') = \{\tau(L) \mid L \in K; \tau \text{ is a fuzzy } K'\text{-substitution}\}$. A family K is *closed* under fuzzy K' -substitution if $\text{Sùb}(K, K') \subseteq K$, and K is *closed under fuzzy substitution*, if K is closed under fuzzy K -substitution. \square

Since we assumed that each family of fuzzy languages is closed under isomorphism, the Sùb -operator is associative, i.e., $\text{Sùb}(K_1, \text{Sùb}(K_2, K_3)) = \text{Sùb}(\text{Sùb}(K_1, K_2), K_3)$; cf. [16, 14].

Taking K and K' equal to families of crisp languages in Definition 4.10 yields the well-known notion of (ordinary, non-fuzzy, crisp) substitution. Then a ONE-substitution is just a homomorphism and an isomorphism (“renaming of symbols”) is a one-to-one SYMBOL-substitution.

Similarly, we define an \mathcal{L} -fuzzy homomorphism $h : \Sigma_1^* \rightarrow \Sigma_2^*$ as an \mathcal{L} -fuzzy ONE_f -substitution. The inverse $h^{-1} : \mathbb{F}(\Sigma_2^*) \rightarrow \mathbb{F}(\Sigma_1^*)$ of such an \mathcal{L} -fuzzy homomorphism is defined by $h^{-1}(L) = \{w \in \Sigma_1^* \mid \mu(h(w); L) > 0\}$ with

$$\mu(x; h^{-1}(L)) = \bigvee \{\mu((x, y); h) \star \mu(y; L) \mid y \in \Sigma_2^*\}. \quad (3)$$

Clearly, h is viewed as a fuzzy relation of which we take the converse to obtain h^{-1} ; cf. (2).

Note that in general for a fuzzy function $f : X \rightarrow Y$ and a fuzzy subset S of Y , we have

$$\begin{aligned} \mu(y; f f^{-1}(S)) &= \bigvee \{\mu(x; f^{-1}(S)) \star \mu((x, y); f) \mid x \in X\} = \\ &= \bigvee \{(\bigvee \{\mu(z; S) \star \mu((x, z); f) \mid z \in Y\}) \star \mu((x, y); f) \mid x \in X\}. \end{aligned}$$

Since f is a function, $\mu((x, z); f) > 0$ and $\mu((x, y); f) > 0$ imply $z = y$. Hence

$$\mu(y; f f^{-1}(S)) = \bigvee \{\mu(y; S) \star \mu((x, y); f) \star \mu((x, y); f) \mid x \in X\} \leq \mu(y; S).$$

Hence $f f^{-1}(S) \subseteq S$, and in case f happens to be a crisp function, we even have equality—i.e., $\mu(y; f f^{-1}(S)) = \mu(y; S)$ —and so $f f^{-1}(S) \sqsubseteq S$ holds. This latter fact we will use in the case of a crisp homomorphism $h : \Sigma^* \rightarrow \text{ONE}$ for which we have $h h^{-1}(S) \sqsubseteq S$; cf. the proofs of Lemma 5.3, Theorem 5.9 and Lemma 5.10.

Proposition 4.11. *The family REG_f is closed under (i) union, (ii) concatenation, (iii) Kleene \star , (iv) fuzzy (regular) substitution, (v) fuzzy homomorphism, and (vi) intersection.*

Proof. The closure properties (i), (ii) and (iii) follow from Definition 4.3 immediately. By a straightforward induction over the structure of a regular fuzzy language one can show closure under fuzzy substitution; cf. Definition 4.10. Since ONE_f is included in REG_f , (iv) implies (v). So it remains to show that REG_f is closed under intersection.

We consider two NFFA's $M_i = (Q_i, \Sigma_i, \delta_i, q_{i0}, F_i)$ ($i = 1, 2$) and we construct a new NFFA $M = (Q_1 \times Q_2, \Sigma_1 \cap \Sigma_2, \delta, (q_{10}, q_{20}), F_1 \times F_2)$ with

$$\begin{aligned} \delta((q_1, q_2), \sigma) &= \{(q'_1, q'_2) \mid q'_1 \in \delta_1(q_1, \sigma), q'_2 \in \delta_2(q_2, \sigma)\} & (\sigma \in \Sigma_1 \cap \Sigma_2), \\ \delta((q_1, q_2), \lambda) &= \{(q'_1, q'_2) \mid q'_1 \in \delta_1(q_1, \lambda), q'_2 \in \delta_2(q_2, \lambda)\} \cup \\ &\cup \{(\{(q'_1, q_2) \mid q'_1 \in \delta_1(q_1, \lambda)\} \cup \{(\{(q_1, q'_2) \mid q'_2 \in \delta_2(q_2, \lambda)\}). \end{aligned}$$

The corresponding degrees of membership are defined by

$$\begin{aligned} \mu((q'_1, q'_2); \delta((q_1, q_2), \sigma)) &= \mu(q'_1; \delta_1(q_1, \sigma)) \wedge \mu(q'_2; \delta_2(q_2, \sigma)), \\ \mu((q'_1, q'_2); \delta((q_1, q_2), \lambda)) &= \mu((q'_1, q_2); \delta((q_1, q_2), \lambda)) = \mu((q_1, q'_2); \delta((q_1, q_2), \lambda)) = 1. \end{aligned}$$

Then $L(M) \doteq L(M_1) \cap L(M_2)$; hence REG_f is closed under intersection. \square

5 Simple Algebraic Structures

We start with a very simple algebraic structure —viz. the fuzzy prequasoid— from which we arrive at more complicated ones such as full AFFL, full substitution-closed AFFL's, etc.; cf. Theorems 5.9, 6.7 and 7.6 below.

Definition 5.1. A normalized family K of fuzzy languages is a *fuzzy prequasoid* if K is closed under fuzzy finite substitution (i.e., $\text{Sûb}(K, \text{FIN}_f) \subseteq K$) and under intersection with regular fuzzy languages. A *fuzzy quasoid* is a fuzzy prequasoid that contains a fuzzy language L_0 such that $c(L_0)$ is infinite. \square

Lemma 5.2. (1) *If K is a fuzzy [pre]quasoid, then $K \supseteq \text{REG}_f$ [$K \supseteq \text{FIN}_f$, respectively].*
(2) *REG_f [FIN_f , respectively] is the smallest fuzzy [pre]quasoid.*
(3) *Let K be a fuzzy prequasoid. If $L \in K$ with $L \subseteq \Sigma^*$ and $c \notin \Sigma$, then $\{c/1\}L \in K$.*

Proof. (1) Let K be a fuzzy prequasoid. Since K is normalized, there is a fuzzy language L over Σ in K that contains a nonempty word x with $\mu(x; L) = 1$. Let a be a symbol occurring in x , and define the fuzzy finite substitutions $\tau : \sigma \mapsto \{\lambda/1, a/1\}$ for each $\sigma \in \Sigma$, and $\varphi : a \mapsto L_F$ where L_F is an arbitrary finite fuzzy language. Then $L_F \doteq \varphi(\tau(L) \cap \{a/1\})$, and hence $L_F \in K$.

If K is a fuzzy quasoid, then K contains an L_0 over Σ_0 such that $c(L_0)$ is infinite. Let R be an arbitrary regular fuzzy language over Σ . Define the fuzzy finite substitution τ by $\tau(\sigma) = \{\lambda/1\} \cup \{\alpha/1 \mid \alpha \in \Sigma\}$ for each $\sigma \in \Sigma_0$. Then $\tau(L_0) \cap R \doteq \{w/1 \mid w \in \Sigma^*\} \cap R \doteq R$, and so R belongs to K .

(2) Since REG_f [FIN_f , respectively] is a [pre]quasoid (cf. Proposition 4.11), statement (2) follows from (1).

(3) Define the crisp finite substitution $\tau : \Sigma^* \rightarrow \text{FIN}$ by $\tau(a) = \{a, ca\}$ and the crisp regular set R by $R = \{c\}\Sigma^*$. Then $\{c/1\}L \doteq \tau(L) \cap R$; hence $\{c/1\}L \in K$. \square

Lemma 5.2 implies that FIN_f is the only fuzzy prequasoid that is not a fuzzy quasoid.

Lemma 5.3. *If a family K of fuzzy languages is closed under fuzzy regular substitution, intersection with regular fuzzy languages and union with regular fuzzy languages, then K is closed under inverse fuzzy homomorphisms.*

Proof. Let (Σ_L, μ_L) be an arbitrary fuzzy language in K where Σ_L is the minimal alphabet of L . Let $h : \Sigma^* \rightarrow \Sigma_L^*$ be a fuzzy homomorphism with $\Sigma = \{\sigma_1, \dots, \sigma_k\}$ and $h(\sigma_i) = w_i$ ($w_i \in \Sigma_L^*$, $1 \leq i \leq k$). We will show that $h^{-1}(L)$ is in K .

First, we assume that L is λ -free. Then we take a new alphabet $\Sigma_0 = \{\sigma'_1, \dots, \sigma'_k\}$ and a crisp λ -free regular substitution τ defined by $\tau(\sigma) = \Sigma_0^* \sigma \Sigma_0^*$ for each σ in Σ_L .

Define L_1 as the finite fuzzy language $L_1 \doteq \{\sigma'_i w_i /_{\mu((\sigma_i, w_i); h)} \mid 1 \leq i \leq k\}$ and the fuzzy language L_2 by $L_2 \doteq \tau(L) \cap L_1^*$. Let h_1 be the crisp homomorphism defined by $h_1(\sigma'_i) = \sigma_i$ ($\sigma'_i \in \Sigma_0$, $\sigma_i \in \Sigma$, and $1 \leq i \leq k$) and $h_1(\alpha) = \lambda$ for each α in Σ_L . Then from the closure properties of K we obtain $L_2 \in K$ and $h_1(L_2) \in K$. It is left to the reader to verify that $h_1(L_2) \doteq h^{-1}(L)$.

When L contains λ , we have $L \doteq (L - \{\lambda\}) \cup \{\lambda\}$ and $h^{-1}(L) \doteq h^{-1}(L - \{\lambda\}) \cup h^{-1}(\lambda)$. Now by the first part of this proof we have $h^{-1}(L - \{\lambda\}) \in K$. If h is λ -free, then $h^{-1}(\lambda) \doteq \{\lambda/1\}$. Otherwise $h^{-1}(\lambda) \doteq \{a /_{\mu((a, \lambda); h)} \mid h(a) = \lambda\}^*$. In either case $h^{-1}(\lambda) \in \text{REG}_f$, and hence $h^{-1}(L) \in K$. \square

Corollary 5.4. *The family REG_f is closed under inverse fuzzy homomorphism.*

Proof. The statement follows from Proposition 4.11 and Lemma 5.3. \square

Next we define three operators on families of fuzzy languages; viz. for each family K of fuzzy languages, let $\Phi_f(K) = \text{Sûb}(K, \text{FIN}_f)$, $\Theta_f(K) = \text{Sûb}(K, \text{ONE}_f)$, and $\Delta_f(K) = \{L \cap R \mid L \in K, R \in \text{REG}_f\}$. Since REG_f is closed under intersection (Proposition 4.11(vi)), and both FIN_f and ONE_f are closed under fuzzy substitution, we have that for $X \in \{\Theta_f, \Delta_f, \Phi_f\}$, X is a closure operator, i.e., (i) X is *extensive*: $K \subseteq X(K)$, (ii) X is *monotonic*: $K_1 \subseteq K_2$ implies $X(K_1) \subseteq X(K_2)$, and (iii) X is *idempotent*: $XX(K) \subseteq X(K)$. Of course, K , K_1 and K_2 denote families of fuzzy languages.

Similarly, let for each family K of fuzzy languages, $\Pi_f(K)$ denote the smallest fuzzy prequasoid that includes K . Clearly, Π_f is a closure operator too.

For each family K of fuzzy languages, we have $\Pi_f(K) = \{\Phi_f, \Delta_f, \Theta_f\}^*(K)$ or even $\Pi_f(K) = \{\Phi_f, \Delta_f\}^*(K)$. But instead of this infinite set of strings over $\{\Phi_f, \Delta_f, \Theta_f\}$ or over $\{\Phi_f, \Delta_f\}$ respectively, a single string suffices; see Proposition 5.5 and Corollary 5.6, respectively.

Proposition 5.5. *For each family K of fuzzy languages, $\Pi_f(K) = \Theta_f \Delta_f \Phi_f(K)$.*

Proof. From the fact that Π_f , Θ_f , Δ_f and Φ_f are closure operators, we infer that $\Pi_f(K) = ((\Delta_f \Phi_f)^* \cup (\Phi_f \Delta_f)^*)(K)$ for each K . Since $\Delta_f \Phi_f(K) \subseteq \Theta_f \Delta_f \Phi_f(K)$ and $\Phi_f \Delta_f(K) \subseteq (\Delta_f \Phi_f)^2(K) \subseteq (\Theta_f \Delta_f \Phi_f)^2(K)$, it remains to show that $(\Theta_f \Delta_f \Phi_f)^2(K) \subseteq \Theta_f \Delta_f \Phi_f(K)$ or, equivalently, that $\Theta_f \Delta_f \Phi_f \Delta_f \Phi_f(K) \subseteq \Theta_f \Delta_f \Phi_f(K)$ for each K .

Suppose $L \in \Theta_f \Delta_f \Phi_f \Delta_f \Phi_f(K)$, i.e., there exist an L_0 in K with $L_0 \subseteq \Sigma_0^*$, fuzzy finite substitutions $\tau_1 : \Sigma_0^* \rightarrow \Sigma_1^*$ and $\tau_2 : \Sigma_1^* \rightarrow \Sigma_2^*$, regular fuzzy languages $R_1 \subseteq \Sigma_1^*$ and $R_2 \subseteq \Sigma_2^*$, and a fuzzy homomorphism $h_1 : \Sigma_2^* \rightarrow \Sigma_3^*$ such that $L \doteq h_1(\tau_2(\tau_1(L_0) \cap R_1) \cap R_2)$.

We will define a fuzzy finite substitution $\tau : \Sigma_0^* \rightarrow \Sigma_4^*$, a regular fuzzy language $R \subseteq \Sigma_4^*$, and a fuzzy homomorphism $h : \Sigma_4^* \rightarrow \Sigma_3^*$ such that $L \doteq h(\tau(L_0) \cap R)$. We assume that $\Sigma_1 \cap \Sigma_2 = \emptyset$. Then we define Σ_4 by $\Sigma_4 = \Sigma_1 \cup \Sigma_2$.

Define crisp homomorphisms $\varphi_i : (\Sigma_1 \cup \Sigma_2) \rightarrow \Sigma_i^*$ by $\varphi_i(\alpha) = \alpha$ for each $\alpha \in \Sigma_i$ and $\varphi_i(\alpha) = \lambda$ otherwise. Let $\tau'_2 : \Sigma_1 \rightarrow (\Sigma_1 \cup \Sigma_2)^*$ be the fuzzy finite substitution defined by $\tau'_2(\alpha) \doteq \{\alpha/1\} \tau_2(\alpha)$ for each α in Σ_1 , let $R \doteq \varphi_1^{-1}(R_1) \cap \varphi_2^{-1}(R_2)$ (which is regular by Corollary 5.4), and $\tau(\sigma) \doteq \tau'_2 \circ \tau_1(\sigma)$ for each $\sigma \in \Sigma_0$, and $h(\alpha) = h_1(\alpha)$ for each $\alpha \in \Sigma_2$ and $h(\alpha) = \lambda$ for each $\alpha \in \Sigma_1$. Then $L \doteq h(\tau(L_0) \cap R)$. \square

Corollary 5.6. *For each family K of fuzzy languages, $\Pi_f(K) = \Phi_f \Delta_f \Phi_f(K)$.* \square

The following algebraic structure is the fuzzy counterpart of the full Abstract Family of Languages or full AFL; cf. [14]. Full substitution-closed AFL's have been investigated in [16].

Definition 5.7. A *full Abstract Family of Fuzzy Languages* or *full AFFL* is a nontrivial family of fuzzy languages closed under union, concatenation, Kleene \star , (possibly erasing) fuzzy homomorphism, inverse fuzzy homomorphism, and intersection with regular fuzzy languages.

A *full substitution-closed AFFL* is a full AFFL closed under fuzzy substitution. \square

The remaining part of this section consists of some elementary results which are straightforward generalizations of their crisp originals (see [14, 16, 2]). First, we consider a characterization of full AFFL in Theorem 5.9 for which we need Lemma 5.3 and the following result.

Lemma 5.8. *A fuzzy prequasoid K is closed under union, concatenation and Kleene \star if and only if K is closed under fuzzy substitution in the regular fuzzy languages.*

Proof. Let K be a fuzzy prequasoid closed under union, concatenation and Kleene \star , and let L_0 be a fuzzy language over Σ_0 from $\text{S}\hat{\text{u}}\text{b}(\text{REG}_f, K)$. Then there is a fuzzy K -substitution $\tau : \Sigma_0 \rightarrow K$ and a regular fuzzy language $R \subseteq \Sigma_0^*$ such that $L_0 \doteq \tau(R)$. By induction on the structure of R we show that $L_0 \in K$.

Basis: If R equals \emptyset , $\{\lambda\}$ or $\{\sigma\}$ ($\sigma \in \Sigma_0$), then clearly $\tau(R) \in K$.

Induction step: Assume that for regular fuzzy languages R_1 and R_2 over Σ_0 , we have that both $\tau(R_1)$ and $\tau(R_2)$ are in K .

If $R \doteq R_1 \cup R_2$, $R \doteq R_1 R_2$ or $R \doteq R_1^*$, we conclude from the induction hypothesis, the closure properties of K and the equalities $\tau(R_1 \cup R_2) \doteq \tau(R_1) \cup \tau(R_2)$, $\tau(R_1 R_2) \doteq \tau(R_1) \tau(R_2)$ and $\tau(R_1^*) \doteq (\tau(R_1))^*$ that $\tau(R) \in K$, which completes the induction.

The converse implication easily follows from substituting fuzzy K -languages into the crisp regular sets $\{a, b\}$, $\{ab\}$ and a^* . \square

Theorem 5.9. *A family K of fuzzy languages is a full AFFL if and only if K is a fuzzy prequasoid closed under fuzzy regular substitution —i.e., $\text{S}\hat{\text{u}}\text{b}(K, \text{REG}_f) \subseteq K$ — and under substitution in the regular fuzzy languages —i.e., $\text{S}\hat{\text{u}}\text{b}(\text{REG}_f, K) \subseteq K$.*

Proof. In view of Lemmas 5.3 and 5.8 it is sufficient to show that $\text{S}\hat{\text{u}}\text{b}(K, \text{REG}_f) \subseteq K$ when K is closed under fuzzy homomorphism, inverse fuzzy homomorphism and intersection with regular fuzzy languages. Note that $\text{S}\hat{\text{u}}\text{b}(K, \text{REG}_f) \subseteq K$ implies closure under fuzzy finite substitution as well.

Let L be a fuzzy K -language over Σ , and let $\tau : \Sigma \rightarrow \text{REG}_f$ be a fuzzy regular substitution with $\tau(\alpha) \subseteq \Sigma_\alpha^*$ for each α in Σ . Define alphabets $\Sigma_0 = \bigcup\{\Sigma_\alpha \mid \alpha \in \Sigma\}$ and $\Sigma_1 = \{\alpha' \mid \alpha \in \Sigma\}$, crisp homomorphisms $h_i : (\Sigma_0 \cup \Sigma_1) \rightarrow \text{ONE}$ ($i = 1, 2$) by $h_1(\alpha') = \alpha$ ($\alpha \in \Sigma_1$), $h_1(\beta) = \lambda$ ($\beta \in \Sigma_0$), $h_2(\alpha') = \lambda$ ($\alpha \in \Sigma_1$), $h_2(\beta) = \beta$ ($\beta \in \Sigma_0$), and the fuzzy language $R \doteq (\bigcup\{\alpha' \tau(\alpha) \mid \alpha \in \Sigma\})^*$ with $\mu(\alpha' x; R) = \mu(x; \tau(\alpha))$ for each $\alpha \in \Sigma$. Then by Lemma 5.2(3) and Proposition 4.11, R is a regular fuzzy language. Now $\tau(L) \doteq h_2(h_1^{-1}(L) \cap R)$ and hence $\tau(L) \in K$. \square

Lemma 5.10. *If K_1 and K_2 are fuzzy prequasoids, then so is $\text{S}\hat{\text{u}}\text{b}(K_1, K_2)$.*

Proof. It is sufficient to show that $\text{S}\hat{\text{u}}\text{b}(K_1, K_2)$ is closed under Φ_f and Δ_f .

First, we have $\Phi_f(\text{S}\hat{\text{u}}\text{b}(K_1, K_2)) = \text{S}\hat{\text{u}}\text{b}(\text{S}\hat{\text{u}}\text{b}(K_1, K_2), \text{FIN}_f) = \text{S}\hat{\text{u}}\text{b}(K_1, \text{S}\hat{\text{u}}\text{b}(K_2, \text{FIN}_f)) = \text{S}\hat{\text{u}}\text{b}(K_1, \Phi_f(K_2)) = \text{S}\hat{\text{u}}\text{b}(K_1, K_2)$ by the associativity of the $\text{S}\hat{\text{u}}\text{b}$ -operation.

Next we will establish the inclusion $\Delta_f(\text{S}\hat{\text{u}}\text{b}(K_1, K_2)) \subseteq \Theta_f(\text{S}\hat{\text{u}}\text{b}(\Delta_f\Phi_f(K_1), \Delta_f(K_2)))$. Since $\Theta_f(\text{S}\hat{\text{u}}\text{b}(\Delta_f\Phi_f(K_1), \Delta_f(K_2))) = \Theta_f(\text{S}\hat{\text{u}}\text{b}(K_1, K_2)) = \text{S}\hat{\text{u}}\text{b}(\text{S}\hat{\text{u}}\text{b}(K_1, K_2), \text{ONE}_f) = \text{S}\hat{\text{u}}\text{b}(K_1, \text{S}\hat{\text{u}}\text{b}(K_2, \text{ONE}_f)) = \text{S}\hat{\text{u}}\text{b}(K_1, \Theta_f(K_2)) = \text{S}\hat{\text{u}}\text{b}(K_1, K_2)$, this inclusion implies the fact that $\Delta_f(\text{S}\hat{\text{u}}\text{b}(K_1, K_2)) \subseteq \text{S}\hat{\text{u}}\text{b}(K_1, K_2)$.

In order to prove the inclusion $\Delta_f(\text{S}\hat{\text{u}}\text{b}(K_1, K_2)) \subseteq \Theta_f(\text{S}\hat{\text{u}}\text{b}(\Delta_f\Phi_f(K_1), \Delta_f(K_2)))$, let L be a fuzzy language over Σ from K_1 , let $\tau : \Sigma^* \rightarrow K_2$ be a fuzzy K_2 -substitution such that $\tau(L) \subseteq \Sigma_1^*$ with $\Sigma_1 \cap \Sigma = \emptyset$, and let R be a regular fuzzy language over Σ_1 . We will prove that $\tau(L) \cap R$ belongs to $\Theta_f(\text{S}\hat{\text{u}}\text{b}(\Delta_f\Phi_f(K_1), \Delta_f(K_2)))$.

We first define the fuzzy substitution τ_2 on Σ^* by $\tau_2(a) \doteq \{a/1\}\tau(a)$ for each a in Σ . Note that by Lemma 5.2(3), τ_2 is a fuzzy K_2 -substitution. Next we define the crisp homomorphism $h : (\Sigma \cup \Sigma_1)^* \rightarrow \text{ONE}$ by $h(a) = \lambda$ for each a in Σ and $h(a) = a$ for each a in Σ_1 . Then $\tau = h \circ \tau_2$ and $\tau(L) \cap R \doteq h\tau_2(L) \cap R \doteq h(\tau_2(L) \cap h^{-1}(R))$ since h is crisp.

Since both R and $h^{-1}(R)$ are regular fuzzy languages (Corollary 5.4), there is according to Proposition 4.9 and Lemma 4.8 an NFFA $M = (Q, \Sigma \cup \Sigma_1, \delta, q_0, \{f\})$ that accepts $h^{-1}(R)$. Let R_0 be defined by

$$R_0 = (L(M) \cap \{\lambda\}) \cup \{(q_0, a_1, q_1) \cdots (q_{m-1}, a_m, q_m) \mid a_i \in \Sigma, q_i \in Q, 1 \leq i \leq m, q_m = f\}.$$

Then R_0 is a crisp regular set (Theorem 2.1, p. 130 in [26] or Lemma 3.2.1 in [14]). Now define for each a in Σ and each p and q in Q the fuzzy language $R(a, p, q)$ by $R(a, p, q) = \{w \mid w \in \Sigma_1^*, q \in \hat{\delta}(p, aw)\}$ with $\mu(w; R(a, p, q)) = \mu(q; \hat{\delta}(p, aw))$. Clearly, $R(a, p, q)$ is a regular fuzzy language by Proposition 4.9, since $R(a, p, q) \doteq L(M(a, p, q))$ where $M(a, p, q)$ is the NFFA defined by $M(a, p, q) = (Q \cup \{q'_0\}, \Sigma_1, \delta', q'_0, \{q\})$ with $\delta' = \delta \cup \{(q'_0, \lambda, q'')/1 \mid q'' \in \delta(p, a)\}$.

Let τ_3 be the fuzzy regular substitution on $(\Sigma \times Q \times Q)^*$ defined by $\tau_3((a, p, q)) \doteq \{a/1\}R(a, p, q)$; cf. Lemma 5.2(3). Then $\tau_3(R_0)$ consists of all words of $h^{-1}(R)$ that do not start with a symbol of Σ_1 . Because $\tau_2(L)$ does not contain words starting with a symbol of Σ_1 , we have $\tau_2(L) \cap h^{-1}(R) \doteq \tau_2(L) \cap \tau_3(R_0)$.

Define the crisp finite substitution τ' on Σ^* by $\tau'(a) = \{a\} \times Q \times Q$ for each a in Σ , and the fuzzy K_2 -substitution τ'' on $(\Sigma \times Q \times Q)^*$ by $\tau''((a, p, q)) \doteq \tau_2(a)$ for each (a, p, q) in $\Sigma \times Q \times Q$. Then $\tau_2 = \tau'' \circ \tau'$, and $\tau_2(L) \cap h^{-1}(R) \doteq \tau''\tau'(L) \cap \tau_3(R_0)$.

Finally, let τ''_3 be the fuzzy K_1 -substitution on $(\Sigma \times Q \times Q)^*$ defined by $\tau''_3((a, p, q)) \doteq \tau''((a, p, q)) \cap \tau_3((a, p, q)) \doteq \{a/1\}\tau(a) \cap \{a/1\}R(a, p, q)$ for each (a, p, q) in $\Sigma \times Q \times Q$.

Then we obtain that $\tau_2(L) \cap h^{-1}(R) \doteq \tau''\tau'(L) \cap \tau_3(R_0) \doteq \tau''_3(\tau'(L) \cap R_0)$. (The actual proof of these two equalities is left as an exercise to the reader.) Consequently, $\tau(L) \cap R \doteq h(\tau''_3(\tau'(L) \cap R_0))$ and hence $\tau(L) \cap R$ belongs to $\Theta_f(\text{S}\hat{\text{u}}\text{b}(\Delta_f\Phi_f(K_1), \Delta_f(K_2)))$. \square

For each family K of fuzzy languages, let $\hat{\mathcal{F}}_f(K)$ denote the smallest full AFFL that includes K . So $\hat{\mathcal{F}}_f$ is a closure operator.

Theorem 5.11. *Let K be a family of fuzzy languages.*

(1) $\text{S}\hat{\text{u}}\text{b}(\text{S}\hat{\text{u}}\text{b}(\text{REG}_f, \Pi_f(K)), \text{REG}_f) = \text{S}\hat{\text{u}}\text{b}(\text{REG}_f, \text{S}\hat{\text{u}}\text{b}(\Pi_f(K), \text{REG}_f))$. *This family of fuzzy languages is a full AFFL that includes K .*

(2) $\hat{\mathcal{F}}_f(K) = \text{S}\hat{\text{u}}\text{b}(\text{S}\hat{\text{u}}\text{b}(\text{REG}_f, \Pi_f(K)), \text{REG}_f) = \text{S}\hat{\text{u}}\text{b}(\text{REG}_f, \text{S}\hat{\text{u}}\text{b}(\Pi_f(K), \text{REG}_f))$.

Proof. (1) The equality follows from the associativity of the $\text{S}\hat{\text{u}}\text{b}$ -operator. Next we show that $\text{S}\hat{\text{u}}\text{b}(\text{S}\hat{\text{u}}\text{b}(\text{REG}_f, \Pi_f(K)), \text{REG}_f)$, abbreviated by $Z(K)$, is a full AFFL that includes K .

By the monotonicity of Π_f , $\text{S}\hat{\text{u}}\text{b}(\text{REG}_f, \cdot)$ and of $\text{S}\hat{\text{u}}\text{b}(\cdot, \text{REG}_f)$, we have $K \subseteq Z(K)$. So it remains to prove that $Z(K)$ is a full AFFL. By the equality of Theorem 5.11(1) and the idempotency of $\text{S}\hat{\text{u}}\text{b}(\text{REG}_f, \cdot)$ and of $\text{S}\hat{\text{u}}\text{b}(\cdot, \text{REG}_f)$ due to Proposition 4.11(iv), it remains to show that $Z(K)$ is a fuzzy prequasoid. However, this follows from Lemmas 5.2 and 5.10.

(2) The inclusion $K \subseteq \hat{\mathcal{F}}_f(K)$, the monotonicity of Z and Theorem 5.9, imply that $Z(K) \subseteq Z\hat{\mathcal{F}}_f(K) = \hat{\mathcal{F}}_f(K)$. As $Z(K)$ is a full AFFL that includes K , we obtain $\hat{\mathcal{F}}_f(K) = Z(K)$. \square

Finally, we turn to full substitution-closed AFFL. Let K_∞ denote the smallest family of fuzzy languages that includes a given family K of fuzzy languages and that is closed under fuzzy substitution.

Theorem 5.12. (1) *If $\text{SYMBOL} \subseteq K$, then*

$$\begin{aligned} K_\infty &= \bigcup_{n=0}^{\infty} \text{SUB}^n(K), & \text{with} \\ \text{SUB}^0(K) &= K, & \text{and} \\ \text{SUB}^{n+1}(K) &= \text{S}\hat{\text{u}}\text{b}(\bigcup_{i=0}^n \text{SUB}^i(K), K), & \text{for each } n \geq 0. \end{aligned}$$

(2) *If K is a fuzzy quasoid, then K_∞ is a full substitution-closed AFFL.*

Proof. (1) Let K_1 denote the family $\bigcup_{n=0}^{\infty} \text{SUB}^n(K)$ for short. Since $\text{SYMBOL} \subseteq K$, we have $\text{SUB}^n(K) \subseteq \text{SUB}^{n+1}(K)$ for each $n \geq 0$. Consequently, $\text{SUB}^{n+1}(K) = \text{S}\hat{\text{u}}\text{b}(\text{SUB}^n(K), K)$ for each $n \geq 0$, and $K = \text{SUB}^0(K) \subseteq K_1$.

The family K_1 is closed under fuzzy K -substitution: viz. let L be a fuzzy language from K_1 —i.e., there is an $i \geq 0$ such that $L \in \text{SUB}^i(K)$ —and let τ be a fuzzy K -substitution. Then $\tau(L) \in \text{SUB}^{i+1}(K)$ and therefore $\tau(L) \in K_1$. Hence $K_\infty \subseteq K_1$.

In order to prove the converse inclusion we show by induction that $\text{SUB}^n(K) \subseteq K_\infty$ for each $n \geq 0$.

Basis: ($n = 0$) $\text{SUB}^0(K) = K \subseteq K_\infty$.

Induction hypothesis: $\text{SUB}^i(K) \subseteq K_\infty$.

Induction step: $\text{SUB}^{i+1}(K) = \text{S}\hat{\text{u}}\text{b}(\text{SUB}^i(K), K) \subseteq \text{S}\hat{\text{u}}\text{b}(K_\infty, K) \subseteq K_\infty$ by the monotonicity of the $\text{S}\hat{\text{u}}\text{b}(\cdot, K)$ -operation, the induction hypothesis and the definition of K_∞ .

Now the inclusions $\text{SUB}^n(K) \subseteq K_\infty$ ($n \geq 0$) imply that $K_1 \subseteq K_\infty$.

(2) By Lemma 5.2 we have $\text{REG}_f \subseteq K \subseteq K_\infty$. Thus K_∞ is closed under $\text{S}\hat{\text{u}}\text{b}(\text{REG}_f, \cdot)$ and under $\text{S}\hat{\text{u}}\text{b}(\cdot, \text{REG}_f)$. According to Theorem 5.9, it suffices to show that K_∞ is a fuzzy prequasoid. However, this can be done using the equality $K_\infty = K_1$ and a straightforward induction in which we use Lemma 5.10. \square

6 More Complicated Algebraic Structures

We first recall the definitions of some generalized fuzzy grammars; they are generalized in the sense that they possess a countably infinite number of rules rather than a finite number. This countable number of rules is restricted in the following way: for each symbol α , the set containing all right-hand sides of rules with left-hand side equal to α forms a fuzzy language that belongs to a given family K of fuzzy languages. This restriction allows us to formulate these grammars in terms of fuzzy K -substitutions. The grammars that have been generalized in this way are: ETOL-system (Definition 6.1), context-free grammar (Definition 6.2), and non-self-embedding context-free grammar (Definition 6.3).

In each case such a family of fuzzy generalized grammars give rise to an algebraic closure operator—viz. H_f , A_f and R_f , respectively—acting on (a slightly restricted class of) families K of fuzzy languages.

Definition 6.1. [5] Let K be a family of fuzzy languages. A *fuzzy K -iteration grammar* $G = (V, \Sigma, U, S)$ consists of an alphabet V , a terminal alphabet Σ ($\Sigma \subseteq V$), an initial symbol S ($S \in V$), and a finite set U of fuzzy K -substitutions over V . The fuzzy language $L(G)$

generated by G is defined by

$$L(G) \doteq U^*(S) \cap \Sigma^* \doteq \bigcup \{ \tau_p(\cdots(\tau_1(S))\cdots) \mid p \geq 0; \tau_i \in U, 1 \leq i \leq p \} \cap \Sigma^*.$$

The family of fuzzy languages generated by fuzzy K -iteration grammars is denoted by $H_f(K)$. \square

Definition 6.2. [8] Let K be a family of fuzzy languages. A *fuzzy context-free K -grammar* G is a fuzzy K -iteration grammar $G = (V, \Sigma, U, S)$ of which each substitution τ from U is a nested fuzzy K -substitution over V ; so $\alpha \in \tau(\alpha)$ for each $\alpha \in V$ and each $\tau \in U$.

The family of fuzzy languages generated by fuzzy context-free K -grammars is denoted by $A_f(K)$. \square

Definition 6.3. [6] Let K be a family of fuzzy languages and let U be a finite set of nested fuzzy K -substitutions over an alphabet V . Then U is called *not self-embedding* if for all $u \in U^*$ and for all α in V , the implication $w_1\alpha w_2 \in u(\alpha) \Rightarrow (w_1 = \lambda \text{ or } w_2 = \lambda)$ holds for all $w_1, w_2 \in V^*$.

A *fuzzy regular K -grammar* $G = (V, \Sigma, U, S)$ is a fuzzy context-free K -grammar where U is a non-self-embedding set of nested fuzzy K -substitutions over V . The family of fuzzy languages generated by fuzzy regular K -grammars is denoted by $R_f(K)$. \square

Example 6.4. (1) When we take K equal to FIN_f , we have $H_f(\text{FIN}_f) = \text{ETOL}_f$ (the family of fuzzy ETOL-languages), $A_f(\text{FIN}_f) = \text{CF}_f$ (the family of fuzzy context-free languages; [22]), and $R_f(\text{FIN}_f) = \text{REG}_f$ (Definition 4.3).

(2) Clearly, we have $\text{CF} \subseteq c(\text{CF}_f)$ where CF is the family of (ordinary, crisp) context-free languages. The converse inclusion does not hold in general.

In order to construct some counterexamples we use (i) the inherent ambiguity of some context-free languages (like, e.g., $\{a^m b^m a^n \mid m, n \geq 1\} \cup \{a^m b^n a^n \mid m, n \geq 1\}$ or $\{a^n b^n c^m d^m \mid m, n \geq 1\} \cup \{a^n b^m c^m d^n \mid m, n \geq 1\}$; cf. Example 3.2(2-3)), and (ii) the structure of the simplest type-00 lattice that is not linearly ordered (cf. Example 2.4(2) in which we have $\xi \vee \eta = 1$); cf. also [25] and Example 4.5.

Consider the type-01 lattice \mathcal{L} of Example 2.4(2) and the \mathcal{L} -fuzzy context-free FIN_f -grammars $G_1 = (V, \Sigma, \{\tau_1\}, S)$ and $G_2 = (V, \Sigma, \{\tau_2\}, S)$ with $V = \{S, A\}$, $\Sigma = \{a, b\}$ and

$$\begin{aligned} \tau_1(\alpha) &\doteq \tau_2(\alpha) \doteq \{\alpha/1\}, & \text{for each } \alpha \text{ in } \Sigma, \\ \tau_1(S) &\doteq \{S/1, Sa/1, Aa/\xi\}, \\ \tau_2(S) &\doteq \{S/1, aS/1, aA/\eta\}, \\ \tau_1(A) &\doteq \{A/1, aAb/1, ab/1\} & \tau_2(A) \doteq \{A/1, bAa/1, ba/1\}. \end{aligned}$$

Then $L(G_1) \doteq L_1$, $L(G_2) \doteq L_2$, $L(G_1) \cup L(G_2) \in \text{CF}_f$, and $c(L(G_1) \cup L(G_2)) = c(L_0)$; for L_0 , L_1 and L_2 we refer to Example 3.2. Note that both G_1 and G_2 are linear context-free and that the support of $L(G_1) \cup L(G_2)$ is an inherently ambiguous, linear context-free language. Since $c(L_0)$ is not (linear) context-free, we have $\text{CF} \subset c(\text{CF}_f)$ and $\text{LCF} \subset c(\text{LCF}_f)$, where LCF [LCF_f , respectively] is the family of [fuzzy] linear context-free languages.

(3) When we restrict ourselves to type-10 lattices \mathcal{L} , then $c(\text{CF}_f) = \text{CF}$. \square

Next we turn to some elementary properties of the families $H_f(K)$, $A_f(K)$ and $R_f(K)$.

Proposition 6.5. [5, 6, 8] (1) *Let K be a family of fuzzy languages closed under union with SYMBOL-languages. If $K \supseteq \text{SYMBOL}$, then $K \subseteq H_f(K)$, $K \subseteq A_f(K)$, and $K \subseteq R_f(K)$.*

(2) *If the family K is a fuzzy prequasoid, then so are $R_f(K)$, $A_f(K)$, and $H_f(K)$.* \square

Now we are ready to consider some algebraic structures that are special cases of full AFFL (Definitions 6.6 and 6.9) and to relate them to these generalized fuzzy grammars (Theorems 6.7, 6.8, 6.10 and 6.11).

Definition 6.6. A family K of fuzzy languages is closed under *iterated fuzzy substitution* if for each fuzzy language L from K with $L \subseteq V^*$ for some alphabet V , and for each finite set U of fuzzy K -substitutions over V , the fuzzy language $U^*(L)$, defined by

$$U^*(L) \doteq \bigcup \{ \tau_p \cdots \tau_1(L) \mid p \geq 0, \tau_i \in U \ (1 \leq i \leq p) \},$$

belongs to K . In case each fuzzy substitution in U is nested, then K is called closed under *nested iterated fuzzy substitution*.

A *full hyper-AFFL* [5] is a full AFFL closed under iterated fuzzy substitution; a *full super-AFFL* [8] is a full AFFL closed under nested iterated fuzzy substitution. \square

For the crisp originals of full substitution-closed AFFL, full super-AFFL and full hyper-AFFL we refer to [16, 14], [18] and [1], respectively. See also [2] for an overview including other algebraic structures weaker than full AFL.

In establishing the following few results Proposition 6.5 played a principal part; cf. [5, 6, 8] for details.

Theorem 6.7. [5, 6, 8] *Let K be a family of fuzzy languages. Then*

- (1) *K is a full substitution-closed AFFL, if and only if K is a fuzzy prequasoid and $R_f(K) = K$.*
- (2) *K is a full super-AFFL, if and only if K is a fuzzy prequasoid and $A_f(K) = K$.*
- (3) *K is a full hyper-AFFL, if and only if K is a fuzzy prequasoid and $H_f(K) = K$.* \square

Theorems 6.7 and 6.8 play the same rôle as Theorems 5.9 and 5.11(1) do with respect to full AFFL's. The proof of Theorem 6.7(1) in [6] heavily relies on Theorem 5.12 above.

Theorem 6.8. [5, 6, 8] *Let K be a family of fuzzy languages. Then*

- (1) *$R_f \Pi_f(K)$ is a full substitution-closed AFFL that includes K .*
- (2) *$A_f \Pi_f(K)$ is a full super-AFFL that includes K .*
- (3) *$H_f \Pi_f(K)$ is a full hyper-AFFL that includes K .* \square

Definition 6.9. Let K be a family of fuzzy languages. By $\hat{\mathcal{R}}_f(K)$ [$\hat{\mathcal{A}}_f(K)$, and $\hat{\mathcal{H}}_f(K)$] we denote the smallest full substitution-closed AFFL, [full super-AFFL, and full hyper-AFFL, respectively] that includes K . \square

Theorem 6.7(3) says that K is a full hyper-AFFL if and only if it is a prequasoid — i.e., $\Pi_f(K) = K$ — and $H_f(K) = K$. Consequently, the smallest full hyper-AFFL $\hat{\mathcal{H}}_f(K)$, that includes a family K , equals $\hat{\mathcal{H}}_f(K) = \bigcup \{ w(K) \mid w \in \{\Pi_f, H_f\}^* \}$ or, equivalently, $\hat{\mathcal{H}}_f(K) = \{\Pi_f, H_f\}^*(K)$. According Theorem 6.10(3) below, this infinite set of strings over $\{\Pi_f, H_f\}$ can be reduced to the single string $H_f \Pi_f$. Obviously, an analogous remark applies to the other full AFFL-structures in Theorems 6.7 and 6.10.

Theorem 6.10. [5, 6, 8] *Let K be a family of fuzzy languages. Then*

- (1) $\hat{\mathcal{R}}_f(K) = R_f \Pi_f(K) = R_f \Theta_f \Delta_f \Phi_f(K)$,
- (2) $\hat{\mathcal{A}}_f(K) = A_f \Pi_f(K) = A_f \Theta_f \Delta_f \Phi_f(K)$, and
- (3) $\hat{\mathcal{H}}_f(K) = H_f \Pi_f(K) = H_f \Theta_f \Delta_f \Phi_f(K)$. \square

Clearly, the latter equalities in Theorem 6.10 have been obtained using Proposition 5.5.

Theorem 6.11. REG_f [CF_f , ETOL_f , respectively] *is the smallest full substitution-closed AFFL [full super-AFFL, full hyper-AFFL].* \square

Each full hyper-AFFL is a full super-AFFL, and each full super-AFFL is a full substitution-closed AFFL. But none of the converse implications hold; cf. Theorem 6.11.

7 An Infinite Sequence of Algebraic Structures

Definition 6.1 is a special instance of a more general fuzzy K -iteration grammar in which the application order of fuzzy K -substitutions is prescribed by a crisp control language over U ; viz.

Definition 7.1. [5] Let Γ be a family of crisp languages, and let K be a family of fuzzy languages. A Γ -controlled fuzzy K -iteration grammar or fuzzy (Γ, K) -iteration grammar is a pair (G, M) that consists of a fuzzy K -iteration grammar $G = (V, \Sigma, U, S)$ and a crisp control language M , i.e., M is a language over U , and $M \in \Gamma$. The fuzzy language $L(G, M)$ generated by (G, M) is defined by

$$L(G, M) \doteq M(S) \cap \Sigma^* \doteq \bigcup \{ \tau_p(\cdots(\tau_1(S))\cdots) \mid p \geq 0; \tau_i \in U, \tau_1 \cdots \tau_p \in M \} \cap \Sigma^*.$$

The family of fuzzy languages generated by fuzzy (Γ, K) -iteration grammars is denoted by both $H_f(\Gamma, K)$ and by $H_{f,\Gamma}(K)$. \square

In comparing Definition 7.1 with Definition 6.1 it is useful to mention the fact that regular control does not extend the generating power of fuzzy K -iteration grammars.

Theorem 7.2. [5] *Let K be a family of fuzzy languages. Then $H_f(\text{REG}, K) = H_f(K)$ holds, provided that $K \supseteq \text{ONE}$.* \square

The number of fuzzy K -substitutions in a $(\Gamma$ -controlled) fuzzy K -iteration grammar can be reduced to two in case the parameters Γ and K satisfy some very simple conditions [5]. In case of a [non-self-embedding] fuzzy context-free K -grammars a reduction to a single, equivalent [non-self-embedding] fuzzy K -substitution is possible [8, 6]. Therefore, providing fuzzy regular or fuzzy context-free K -grammars with a control language, that prescribes the application order of the [non-self-embedding] fuzzy K -substitutions, will probably not result into an interesting topic.

In order to give some elementary properties of $H_{f,\Gamma}(K)$ we need the following concepts.

Definition 7.3. A crisp family Γ is closed under *left marking* [*right marking*] if for each language L in Γ with $L \subseteq \Sigma^*$ for some Σ , and for each c not in Σ , the language $\{c\}L$ [$L\{c\}$, respectively] belongs to Γ . And Γ is closed under *full marking* if Γ is closed under both left and right marking. \square

Proposition 7.4. [5] (1) *Let Γ be a crisp family closed under right marking, and let K be a family of fuzzy languages with $K \supseteq \text{ONE}$. Then $\Gamma \subseteq H_{f,\Gamma}(K)$ and $K \subseteq H_{f,\Gamma}(K)$.*

(2) *Let Γ be a crisp family closed under (i) left or right marking, (ii) union or concatenation, and (iii) Kleene \star . If K is a family of fuzzy languages with $K \supseteq \text{SYMBOL}$, then $H_f(K) \subseteq H_{f,\Gamma}(K)$.*

(3) *Let Γ be a crisp family closed under full marking. If K is a fuzzy prequasoid, then so is $H_{f,\Gamma}(K)$.* \square

It is useful to compare Proposition 7.4(1) and (3) with the corresponding statements in Proposition 6.5(1) and (2), respectively.

Next we generalize the notion of iterated fuzzy substitution to Γ -controlled iterated fuzzy substitution where Γ is a family of crisp languages.

Definition 7.5. Let Γ be a family of crisp languages. A family K of fuzzy languages is closed under Γ -controlled iterated fuzzy substitution, if for each fuzzy language L from K with

$L \subseteq V^*$ for some alphabet V , for each finite set U of fuzzy K -substitutions over V , and for each crisp language M over U from the family Γ , the fuzzy language $M(L)$, defined by

$$M(L) \doteq \bigcup \{ \tau_p \cdots \tau_1(L) \mid p \geq 0, \tau_i \in U \ (1 \leq i \leq p), \tau_1 \cdots \tau_p \in M \},$$

belongs to K ; cf. Definition 6.6. A *full Γ -hyper-AFFL* is a full AFFL closed under Γ -controlled iterated fuzzy substitution.

For each family K , let $\hat{H}_{f,\Gamma}(K)$ be the smallest full Γ -hyper-AFFL that includes K . \square

Theorem 7.6. *Let the crisp family Γ be a full substitution-closed AFL. Then a family K of fuzzy languages is a full Γ -hyper-AFFL if and only if K is a fuzzy prequasoid and $H_{f,\Gamma}(K) = K$.*

Proof. Suppose K is a full Γ -hyper-AFFL. By Theorem 5.9, K is a fuzzy prequasoid; so it remains to show that $H_{f,\Gamma}(K) \subseteq K$, as the converse inclusion follows from Proposition 7.4(1).

Let (G, M) be an arbitrary Γ -controlled fuzzy K -iteration grammar. So $M \in \Gamma$ and $G = (V, \Sigma, U, S)$. Because K is a full Γ -hyper-AFFL, the fuzzy languages $\{S/1\}$, $M(\{S/1\})$ and $M(\{S/1\}) \cap \Sigma^*$ all belong to the family K . But the latter fuzzy language equals $L(G, M)$. Hence $L(G, M) \in K$ and $H_{f,\Gamma}(K) \subseteq K$.

Conversely, let K be a fuzzy prequasoid that satisfies $H_{f,\Gamma}(K) = K$. First, we show that K is closed under Γ -controlled iterated fuzzy substitution.

Let L_0 be an arbitrary fuzzy language in K with $L_0 \subseteq V^*$ for some alphabet V , and let U be a finite set of fuzzy K -substitutions over V and let $M \subseteq U^*$ be a crisp language from Γ . Consider the Γ -controlled fuzzy K -iteration grammar $(G, \{\tau\}M)$ with $G = (V \cup \{S\}, V, U \cup \{\tau\}, S)$, $S \notin V$, $\tau \notin U$ and $\tau(S) \doteq L_0$ and $\tau(\alpha) \doteq \{\alpha/1\}$ for each α in V .

Then $L(G, \{\tau\}M) \doteq M(L_0)$, $L(G, \{\tau\}M) \in H_{f,\Gamma}(K) = K$, and hence $M(L_0) \in K$, i.e., K is closed under Γ -controlled iterated fuzzy substitution.

As K is a fuzzy prequasoid, we have $\text{FIN}_f \subseteq K$ and thus $\text{REG}_f \subseteq \text{ETOL}_f = H_f(\text{FIN}_f) = H_{f,\text{REG}}(\text{FIN}_f) \subseteq H_{f,\Gamma}(K) = K$ by Example 6.4(2) and Theorem 7.2. But $K \subseteq R_f(K) \subseteq H_f(K) = H_{f,\text{REG}}(K) \subseteq H_{f,\Gamma}(K) = K$ according to Definitions 6.1 and 6.3, Theorem 7.2 and the fact that $\Gamma \supseteq \text{REG}$. So $R_f(K) = K$ and by Theorem 5.9 or 6.7(1), K is a full AFFL. \square

Theorem 7.6 is the analogue of Theorem 6.7 as Theorem 7.8(2), (3) and (4) are of Theorems 6.8, 6.10 and 6.11, respectively. However, to establish Theorem 7.8 we need the main result from [5], viz.

Theorem 7.7. (1) *Let Γ_1 and Γ_2 be families of crisp languages and let Γ_2 be closed under full marking, union or concatenation, and Kleene \star . If K is a family of fuzzy languages with $K \supseteq \text{ALPHA}$, then $H_f(\Gamma_1, H_f(\Gamma_2, K)) \subseteq H_f(\text{S}\hat{\cup}(\Gamma_1, \Gamma_2), K)$.*

(2) *Let Γ be a family of crisp languages closed under full marking and under substitution that satisfies $\Gamma \supseteq \text{REG}$. If K is a family of fuzzy languages with $K \supseteq \text{ALPHA} \cup \text{ONE}$, then $H_f(\Gamma, H_f(\Gamma, K)) = H_f(\Gamma, K)$.*

(3) *Let Γ be a family of crisp languages closed under full marking, union, concatenation, and Kleene \star . If K is a family of fuzzy languages with $K \supseteq \text{ALPHA} \cup \text{ONE}$, then $H_f(H_f(\Gamma, K)) = H_f(\Gamma, K)$. \square*

Theorem 7.8. *Let the crisp family Γ be a full substitution-closed AFL, and let K be a family of fuzzy languages.*

(1) *Each full Γ -hyper-AFFL is a full hyper-AFFL.*

(2) *$H_{f,\Gamma}\Pi_f(K)$ is a full Γ -hyper-AFFL that includes K .*

$$(3) \hat{\mathcal{H}}_{f,\Gamma}(K) = H_{f,\Gamma}\Pi_f(K) = H_{f,\Gamma}\Theta_f\Delta_f\Phi_f(K).$$

(4) $H_{f,\Gamma}(\text{FIN}_f)$ is the smallest full Γ -hyper-AFFL.

Proof. (1) Clearly, by Theorems 6.7(3) and 7.6 it is sufficient to show that $H_{f,\Gamma}(K) = K$ implies $H_f(K) = K$. Since $\Gamma \supseteq \text{REG}$, we have by Propositions 6.5(1) and 7.4(2): $K \subseteq H_f(K) \subseteq H_{f,\Gamma}(K) = K$. Hence $H_f(K) = K$.

(2) This result follows from Proposition 7.4(3), Theorems 7.6 and 7.7(2).

(3) By the inclusion $K \subseteq \hat{\mathcal{H}}_{f,\Gamma}(K)$ and the monotonicity of both $H_{f,\Gamma}$ and Π_f , we have $H_{f,\Gamma}\Pi_f(K) \subseteq H_{f,\Gamma}\Pi_f\hat{\mathcal{H}}_{f,\Gamma}(K)$. According to Theorem 7.6, this yields $H_{f,\Gamma}\Pi_f(K) \subseteq \hat{\mathcal{H}}_{f,\Gamma}(K)$. Now Theorem 7.8(2) implies that $H_{f,\Gamma}\Pi_f(K)$ is a full Γ -hyper-AFFL that includes K . Hence we obtain that $\hat{\mathcal{H}}_{f,\Gamma}(K) = H_{f,\Gamma}\Pi_f(K)$.

(4) This statement follows from Theorem 7.8(3) and Lemma 5.2(2) (FIN_f is the smallest fuzzy prequasoid). \square

Comparing Theorem 7.8(3) and the obvious equality $\hat{\mathcal{H}}_{f,\Gamma}(K) = \{H_{f,\Gamma}, \Pi_f\}^*(K)$, shows that the single string $H_{f,\Gamma}\Pi_f$ suffices rather than the countably infinite set $\{H_{f,\Gamma}, \Pi_f\}^*$.

The free parameter Γ allows us to proceed inductively over the crisp family of control languages, yielding an infinite sequence of signatures (types/classes of algebras).

Theorem 7.9. *Let K be a \mathcal{L} -fuzzy prequasoid, and let $Q_0 = \text{REG}$ and $Q_{i+1} = H_f(c(Q_i), K)$ for each $i \geq 0$. Then for each $i \geq 0$, Q_j is a full $c(Q_i)$ -hyper-AFFL provided that $j > i$.*

Proof. A straightforward inductive argument on i —applying Theorem 7.2, Propositions 7.4(1) and 7.4(3), and Theorem 7.7(3)— yields the following facts:

- (7.9-i) Q_i is a full hyper-AFFL for each $i \geq 1$, and
- (7.9-ii) $Q_i \subseteq Q_j$ provided $j \geq i$.

Using these facts we will prove by induction on i that Q_j is a full $c(Q_i)$ -hyper-AFFL for each j with $0 \leq i < j$.

Basis: ($i = 0$). We have to show that for each $j \geq 1$, Q_j is a full $c(Q_0)$ -hyper-AFFL. Since $Q_0 = \text{REG}$ and each Q_j is a full $c(\text{REG})$ -hyper-AFFL if and only if Q_j is a full hyper-AFFL (Theorem 7.8(1)), the statement follows from (7.9-i).

Induction hypothesis: Assume that for each $j > i$, Q_j is a full $c(Q_i)$ -hyper-AFFL.

Induction step: We have to show that each family Q_j with $j > i + 1$ is a full $c(Q_{i+1})$ -hyper-AFFL.

Consider an arbitrary Q_j with $j > i + 1$; then $Q_j = H_f(c(Q_{j-1}), K)$. As $j - 1 > i$, the induction hypothesis implies that Q_{j-1} is a full $c(Q_i)$ -hyper-AFFL. Now by Theorem 7.8(1) and Proposition 7.4(3), Q_j is a fuzzy prequasoid.

So it remains to show that $H_f(c(Q_{i+1}), Q_j) \subseteq Q_j$, since the converse inclusion follows from Proposition 7.4(2) and (7.9-i).

From the definition of Q_j and Theorem 7.7(1) respectively, we obtain

$$H_f(c(Q_{i+1}), Q_j) = H_f(c(Q_{i+1}), H_f(c(Q_{j-1}), K)) \subseteq H_f(c(\text{S}\hat{\text{u}}\text{b}(Q_{i+1}, Q_{j-1}), K).$$

We already remarked that the induction hypothesis implies that Q_{j-1} is a full $c(Q_i)$ -hyper-AFFL. By Theorem 7.8(1), Q_{j-1} is a full hyper-AFFL and so Q_{j-1} is closed under fuzzy substitution. Consequently, $c(Q_{j-1})$ is closed under (ordinary, crisp) substitution. As $j - 1 \geq i + 1$, we have $Q_{i+1} \subseteq Q_{j-1}$ by (7.9-ii), and consequently, $c(\text{S}\hat{\text{u}}\text{b}(Q_{i+1}, Q_{j-1})) \subseteq c(Q_{j-1})$. Hence we have $H_f(c(Q_{i+1}), Q_j) \subseteq H_f(c(\text{S}\hat{\text{u}}\text{b}(Q_{i+1}, Q_{j-1}), K) \subseteq H_f(c(Q_{j-1}), K) = Q_j$, which completes the induction. \square

Note that the statement of Theorem 7.9 still contains two free parameters, viz. (i) the fuzzy prequasoid K , and (ii) the type-00 lattice \mathcal{L} . To make the latter dependency explicit, we wrote “ \mathcal{L} -fuzzy” rather than “fuzzy” in Theorem 7.9.

By fixing K and restricting \mathcal{L} we are able to establish the existence of a countably infinite sequence of full AFFL-structures: see Theorem 7.12, the proof of which relies on Theorem 7.9 and the following two results.

In Theorem 7.10 $H(\Gamma, K)$ denotes the family of languages $L(G, M)$ generated by (ordinary, crisp) (Γ, K) -iteration grammars (G, M) , i.e., all substitutions involved in G are crisp K -substitutions; cf. [1].

Theorem 7.10. [11, 12] *Let $K_0 = \text{REG}$ and $K_{i+1} = H(K_i, \text{FIN})$ for each $i \geq 0$. Then $\{K_i\}_{i \geq 1}$ is an infinite hierarchy of full hyper-AFL's, i.e.,*

- for each $i \geq 1$, K_i is a full hyper-AFL, and
- for each $i \geq 1$, K_i is properly included in K_{i+1} : $K_i \subset K_{i+1}$. □

Corollary 7.11. *Let \mathcal{L} be an arbitrary type-10 lattice, and let $\{F_i\}_{i \geq 1}$ be the sequence of families of \mathcal{L} -fuzzy languages defined by $F_0 = \text{REG}_f$ and $F_{i+1} = H_f(c(F_i), \text{FIN}_f)$ for each $i \geq 0$. Then $\{F_i\}_{i \geq 1}$ is an infinite hierarchy of full hyper-AFFL's, i.e.,*

- for each $i \geq 1$, F_i is a full hyper-AFFL, and
- for each $i \geq 1$, F_i is properly included in F_{i+1} : $F_i \subset F_{i+1}$.

Proof. First, we show by induction that for each $i \geq 0$, $c(F_i) = K_i$ where $\{K_i\}_{i \geq 1}$ is as in Theorem 7.10.

Basis: ($i = 0$). $c(F_0) = K_0$ follows from Lemma 4.3(2).

Induction hypothesis: Assume $c(F_m) = K_m$.

Induction step: In order to prove $c(F_{m+1}) = K_{m+1}$, we first remark that $K_{m+1} \subseteq c(F_{m+1})$. Hence it remains to show that $c(F_{m+1}) \subseteq K_{m+1}$.

So let L_0 be an arbitrary element of $c(F_{m+1})$, i.e., $L_0 = c(L_f)$ for some $L_f \in F_{m+1}$. Thus there exists a fuzzy $(c(F_m), \text{FIN}_f)$ -iteration grammar (G, M) with $G = (V, \Sigma, U, S)$ such that $L(G, M) \doteq L_f$. By the induction hypothesis, (G, M) is a fuzzy (K_m, FIN_f) -iteration grammar. Next we will construct an equivalent (K_m, FIN) -iteration grammar (G', M) by $G' = (V, \Sigma, U', S)$, $U' = \{\tau' \mid \tau \in U\}$ and for each α in V and each τ in U , we define $\tau'(\alpha) = c(\tau(\alpha))$.

Since \mathcal{L} is linearly ordered, the max-operation applies and as $a \leq \max\{a, b\} = \max\{b, a\}$ for all $a, b \in \mathcal{L}$, we have that the crisp language $L(G', M)$ equals $c(L_f)$. Consequently, we have $L_0 \in H(K_m, \text{FIN})$ or, equivalently, $L_0 \in K_{m+1}$, which completes the induction.

Now the statement follows from Theorems 7.7(3) and 7.10. □

Finally, we are ready for the main result.

Theorem 7.12. *Let \mathcal{L} be an arbitrary type-10 lattice and consider the following families of \mathcal{L} -fuzzy languages: $F_0 = \text{REG}_f$ and $F_{m+1} = H_f(c(F_m), \text{FIN}_f)$ for $m \geq 0$. Let \mathcal{C}_m be the class of all full $c(F_m)$ -hyper-AFFL's. Then for each $m \geq 1$,*

- (1) *the class \mathcal{C}_m is a proper superset of \mathcal{C}_{m+1} : $\mathcal{C}_m \supset \mathcal{C}_{m+1}$,*
- (2) *the class \mathcal{C}_m contains an infinite hierarchy of full $c(F_m)$ -AFFL's, i.e., a countably infinite chain of families of fuzzy language $F_{m,n}$ ($n \geq 1$) such that*
 - (i) *each $F_{m,n}$ is a full $c(F_m)$ -AFFL, and*
 - (ii) *for each $n \geq 1$, $F_{m,n}$ is properly included in the next one: $F_{m,n} \subset F_{m,n+1}$.*

Proof. (1) The statement follows from Corollary 7.11 and Theorems 7.9 (with $K = \text{FIN}_f$) and 7.8(4).

(2) For fixed m ($m \geq 1$), we define $\{F_{m,n}\}_{n \geq 1}$ by $F_{m,n} = F_{m+n}$ for each $n \geq 1$. By Corollary 7.11 and Theorem 7.9 this is an infinite hierarchy of full $c(F_m)$ -hyper-AFFL's. \square

8 Concluding Remarks

In Sections 4–5 we showed that some basic results for crisp language families (like prequasoid and full AFL) can be generalized to their fuzzy analogues (fuzzy prequasoid and full AFFL, respectively), provided the operations on fuzzy languages have been defined appropriately (Section 3). Then in Section 6 we surveyed some results on full substitution-closed AFFL's, full super-AFFL's and full hyper-AFFL's from [5, 6, 8]. In Section 7 we extended this finite chain of algebraic structures to a countably infinite sequence of full AFFL-structures, each of which possesses properties (Theorem 7.8) similar to those of the members of the initial, finite sequence (Theorems 6.8, 6.10 and 6.11). And each new class of full AFFL-structures in this sequence is nontrivial in the sense that it contains a countably infinite hierarchy (Theorem 7.12).

Note that this latter conclusion has only been proved for fuzzy languages of which the codomain \mathcal{L} of the membership function is linearly ordered (a type-10 lattice; Section 2). Whether this result can be generalized to arbitrary type-00 lattices is an open question, but its answer is probably negative. The approach in Section 7, i.e., deriving Corollary 7.11 from Theorem 7.10, will not work as we will show. More precisely: if \mathcal{L} is a type-01 lattice, K_f is a family of \mathcal{L} -fuzzy languages and K is its crisp counterpart, then —apart from a few trivial exceptions (viz. K_f equals FIN_f or ALPHA_f ; cf. Lemma 4.3)— in general K seems to be a proper subset of $c(K_f)$: $K \subset c(K_f)$; cf. Example 6.4(2) for the case $K = \text{CF}$.

Proper inclusions of this kind prevent us to apply an argument as in the proof of Corollary 7.11 in case \mathcal{L} is a type-00 lattice that is not linearly ordered.

Note that the question whether $c(\text{REG}_f) = \text{REG}$ in case \mathcal{L} is a type-00 lattice, is still open; cf. Lemma 4.3(2).

A “crisp version” of Theorem 7.12 has been established in [7]: in that case the smallest elements (Theorem 7.8(4)) are subfamilies of the family of context-sensitive languages CS; see [7] for details.

Another topic for further investigation is the limit family of fuzzy languages F_ω , defined by $F_\omega = \bigcup_{n \geq 0} F_n$ (cf. Theorem 7.12). As its crisp counterpart $K_\omega = \bigcup_{n \geq 0} K_n$ (Theorem 7.10), it possesses closure properties, even stronger than those of full $c(F_n)$ -hyper-AFFL, viz. $F_\omega = H_f(c(F_\omega), F_\omega)$. With respect to K_ω we know that $K_\omega \subset \text{CS}$ [7, 11, 12], but where the position of F_ω is in the extended “fuzzified” Chomsky hierarchy, is still open.

References

1. P.R.J. Asveld, Controlled iteration grammars and full hyper-AFL's, *Inform. Contr.* **34** (1977) 248–269.
2. P.R.J. Asveld, An algebraic approach to incomparable families of formal languages, pp. 455–475 in: G. Rozenberg & A. Salomaa (eds.), *Lindenmayer Systems — Impacts on Theoretical Computer Science, Computer Graphics, and Developmental Biology* (1992), Springer, Berlin, etc.
3. P.R.J. Asveld, Towards robustness in parsing — Fuzzifying context-free language recognition, pp. 443–453 in: J. Dassow, G. Rozenberg & A. Salomaa (eds.), *Developments in*

- Language Theory II — At the Crossroads of Mathematics, Computer Science and Biology*, World Scientific (1996), Singapore.
4. P.R.J. Asveld, A fuzzy approach to erroneous inputs in context-free language recognition, pp. 14–25 in: *Proc. 4th Internat. Workshop on Parsing Technologies IWPT'95* (1995), Prague / Karlovy Vary, Czech Republic.
 5. P.R.J. Asveld, Controlled fuzzy parallel rewriting, pp. 49–70 in: Gh. Păun & A. Salomaa (eds.), *New Trends in Formal Languages — Control, Cooperation, and Combinatorics* (1997), Lect. Notes in Comp. Sci. **1218**, Springer, Berlin, etc.
 6. P.R.J. Asveld, The non-self-embedding property for generalized fuzzy context-free grammars, *Publ. Math. Debrecen* **54 Suppl.** (1999) 553–573.
 7. P.R.J. Asveld, An infinite sequence of full AFL-structures, each of which possesses an infinite hierarchy, Chapter 15 (pp. 175–186) in: C. Martin-Vide & V. Mitrana (eds.): *Where Mathematics, Computer Science and Biology Meet* (2001), Kluwer, Dordrecht, the Netherlands.
 8. P.R.J. Asveld, Fuzzy context-free languages — Part I: Generalized fuzzy context-free grammars (2000), CTIT Tech. Rep. No. 00-03, Enschede, the Netherlands.
 9. P.R.J. Asveld, Fuzzy context-free languages — Part II: Recognition and parsing algorithms (2000), CTIT Techn. Rep. No. 00-25, Enschede, the Netherlands.
 10. R.V. Book, S. Even, S.A. Greibach & G. Ott, Ambiguity in graphs and expressions, *IEEE Trans. on Comp.* **C-20** (1971) 149–153.
 11. J. Engelfriet, Three hierarchies of transducers, *Math. Syst. Th.* **15** (1982) 95–125.
 12. J. Engelfriet, Hierarchies of hyper-AFL's, *J. Comp. Syst. Sci.* **30** (1985) 86–115.
 13. G. Gerla, Fuzzy grammars and recursively enumerable fuzzy languages, *Inform. Sci.* **60** (1992) 137–143.
 14. S. Ginsburg, *Algebraic and Automata-Theoretic Properties of Formal Languages* (1975), North-Holland, Amsterdam.
 15. S. Ginsburg, S.A. Greibach & J.E. Hopcroft, *Studies in Abstract Families of Languages*, Memoirs of the Amer. Math. Soc., No. 87 (1969).
 16. S. Ginsburg & E.H. Spanier, Substitution in families of languages, *Inform. Sci.* **2** (1970) 83–110.
 17. J.A. Goguen, *L*-fuzzy sets, *J. Math. Analysis Appl.* **18** (1967) 145–174.
 18. S.A. Greibach, Full AFL's and nested iterated substitution, *Inform. Contr.* **16** (1970) 7–35.
 19. M.A. Harrison, *Introduction to Formal Language Theory* (1978), Addison-Wesley, Reading, Mass.
 20. J.E. Hopcroft & J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation* (1979), Addison-Wesley, Reading, Mass.
 21. H.H. Kim, M. Mizumoto, J. Toyoda & K. Tanaka, *L*-fuzzy grammars, *Inform. Sci.* **8** (1975) 123–140.
 22. E.T. Lee & L.A. Zadeh, Note on fuzzy languages, *Inform. Sci.* **1** (1969) 421–434.

23. D.S. Malik & J.N. Mordeson, On fuzzy regular languages, *Inform. Sci.* **88** (1996) 263–273.
24. A. Mateescu, A. Salomaa, K. Salomaa & S. Yu, Lexical analysis with a simple finite fuzzy automaton model, *J. of Universal Comp. Sci.* **1** (1995) 292–311.
25. M. Mizumoto, J. Toyoda & K. Tanaka, Examples of formal grammars with weights, *Inform. Process. Lett.* **2** (1973) 74–78.
26. A. Salomaa, *Formal Languages* (1973), Academic Press, New York.
27. E.S. Santos, Fuzzy automata and languages, *Inform. Sci.* **10** (1976) 193–197.
28. E.S. Santos, Regular fuzzy expressions, pp. 169–175 in M.M. Gupta, G.N. Saridis & B.R. Gaines (eds.), *Fuzzy Automata and Decision Processes* (1977), North-Holland, New York – Amsterdam – Oxford.
29. T.A. Sudkamp, *Languages and Machines — An Introduction to the Theory of Computer Science*, Second edition (1997), Addison-Wesley, Reading, Mass.
30. W. Wechler, *The Concept of Fuzziness in Automata and Language Theory* (1978), Akademie-Verlag, Berlin.
31. W. Wechler, A note on closure properties of R -fuzzy events, *Found. Contr. Engrg.* **3** (1978) 191–202.