Fuzzy Context-Free Languages — Part 1: Generalized Fuzzy Context-Free Grammars

Peter R.J. Asveld

Department of Computer Science, Twente University of Technology
P.O. Box 217, 7500 AE Enschede, the Netherlands

Abstract

Motivated by aspects of robustness in parsing a context-free language, we study generalized fuzzy context-free grammars. These fuzzy context-free K-grammars provide a general framework to describe correctly as well as erroneously derived sentences by a single generating mechanism. They model the situation of making a finite choice out of an infinity of possible grammatical errors during each context-free derivation step. Formally, a fuzzy context-free K-grammar is a fuzzy context-free grammar with a countable rather than a finite number of rules satisfying the following condition: 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. We investigate the generating power of fuzzy context-free K-grammars, and we show that under minor assumptions on the parameter K, the family of languages generated by fuzzy context-free K-grammars possesses closure properties very similar to those of the family of ordinary context-free languages.

Key words: formal language, fuzzy context-free grammar, grammatical error, algebraic closure property

1 Introduction

Usually there are many different reasons to call a specific parser for a language more or less robust. One obvious aspect of robustness in parsing is the requirement that, given a slightly incorrect input, the parser still behaves as were its input flawless. In an ideal setting the parser could even report which corrections it made in the original input in order to produce the desired output. However, in making this intuitive description of robustness more formal, we encounter a few serious problems. First of all, what is a tiny mistake and what is a big irreparable error in the input of a parser? In the usual formal

Email address: `infprja@cs.utwente.nl` (Peter R.J. Asveld).

way of dealing with languages and parsing we are unable to distinguish these two types of errors at all.

In that formal approach we have the following situation: Given a language L_0 over an alphabet Σ, we construct a finite description of L_0 by means of a context-free grammar G such that the language $L(G)$ generated by G satisfies $L(G) = L_0$, and finally, we develop a parser M based on G. Of course, the domain of M equals $\Sigma^\star$; on input strings x from $L(G)$ the parser accepts (and delivers some syntactic description of x according to G), whereas strings from $\Sigma^\star - L(G)$ are simply rejected as not belonging to L_0. So the least we demand from a parser M for a language L_0 is, that M recognizes the language L_0, i.e., that M is able to compute the characteristic function $\mu_{L_0} : \Sigma^\star \to \{0,1\}$ of L_0 defined by $\mu_{L_0}(x) \ = \ \textbf{if } x \in L_0 \textbf{ then } 1 \textbf{ else } 0$.

Thus an input x is either correct (in case $\mu_{L_0}(x) = 1$) or incorrect (when $\mu_{L_0}(x) = 0$) and there is no room for subtleties like a distinction between a "tiny mistake" and a "capital blunder". Clearly, a way out is to demolish the sharp boundary between *in* (i.e., $\mu_{L_0}(x) = 1$) and *out* (i.e., $\mu_{L_0}(x) = 0$) the language L_0. This leads to the concept of fuzzy language [20], being a language L_0 over an alphabet Σ provided with a membership function $\mu_{L_0} : \Sigma^\star \to [0,1]$. Note that the set $\{0,1\}$ with two elements has been replaced by the real closed interval $[0,1]$ and, consequently, now $\mu_{L_0}(x)$ can take any real value in between 0 and 1. Thus this approach allows for describing "tiny mistakes" x with $\Delta \leq \mu_{L_0}(x) < 1$ as well as "capital blunders" x with $0 < \mu_{L_0}(x) \leq \delta$ with respect to L_0, once we made an appropriate choice for the thresholds Δ and δ. However, in order to model the accumulation of grammatical errors —i.e., making an error twice is worse than making it once— we will use in this paper a completely distributive complete lattice provided with an additional operation [6–8] as codomain for membership functions rather than the special case of the real closed interval [0,1]; cf. Sections 2, 3 and 4 for details.

The second question we address is the type of errors we allow in the input of the parser and the way we generate these errors. In view of the discussion above, a fuzzy context-free grammar [20] is a natural choice to generate a fuzzy context-free language. Such a fuzzy context-free grammar G generates, apart from the usual "correct strings" x (with $\mu_{L(G)}(x) = 1$), some "incorrect strings" x' (with $0 < \mu_{L(G)}(x') < 1$) due to grammatical errors as well. So erroneous inputs to a parser are assumed to be generated by grammatical errors, and in order to obtain these grammatical errors we extend the original context-free grammar with some additional rules resulting in a fuzzy context-free grammar (Section 4).

But we will run to an extreme by using fuzzy context-free K-grammars (Section 5), i.e., fuzzy context-free grammars with a countable rather than a finite number of grammar rules. This concept models the feature that, in general, there is an infinite number of ways in which we may apply a given grammar rule erroneously. The notion of fuzzy context-free K-grammar happens to be a general way to describe context-free languages with both correct as well as erroneous sentences generated by a single grammatical device (Sections 5, 6 and 7). Provided that the parameter K satisfies some minor assumptions, the family of languages generated by these fuzzy context-free K-grammars shares many of the interesting algebraic closure properties that the family of (ordinary) context-free languages possesses; cf. Sections 8 and 9. Finally, Section 10 contains some discussion and a few concluding remarks.

The third problem related to erroneous inputs of parsers, is the concept of robustness in parsing and recognizing (fuzzy) context-free languages. However, this topic is postponed to the companion [9] of the present paper.

Of these two papers, the present one (Part 1) deals with rudiments (Sections 1–4) and theoretical issues (Sections 5–9). So readers interested in more practical aspects, like recognition and parsing, are referred to Sections 1–4 of Part 1 and then to Part 2.

The results in this paper and its companion [9] are extensions of simpler ones announced in [4,5]. The present generalizations have been suggested by related work in [7] on a restricted type of fuzzy context-free K-grammar and in [6] on parallel fuzzy rewriting systems.

Finally, we emphasize that we use fuzzy languages purely at a *syntactical* level, i.e., for describing the quality of a string x generated by a fuzzy context-free grammar (viz. x is completely correct / a tiny mistake / a capital blunder / completely incorrect). Note that this approach differs considerably from modeling "vagueness" or "uncertainty" in natural language fragments, which occurs at a purely *semantical* level (viz. by translating a sentence from a non-fuzzy context-free language to a formula in first-order fuzzy logic or to an element of a domain defined in terms of fuzzy sets); cf. [19] for many papers on this latter subject.

2 Preliminaries

For all unexplained terminology and notation on formal languages and grammars we refer to standard texts like [1,15,16]. We also need some rudiments of lattice theory which can be found in many books on algebra; see also [3]. Before we turn to fuzzy languages we fix some notation with respect to ordinary (or crisp) formal languages.

An *alphabet* Σ is a finite set of symbols. A *word* or *string* over Σ is a finite sequence of symbols from Σ. The *empty word* is denoted by λ. For each alphabet Σ, $\Sigma^\star$ [Σ^+, respectively] is the set of all [nonempty] words over Σ. Let $|w|$ denote the length of the word w; so $|\lambda| = 0$, and for all w in Σ^+: if $w = ax$ with $a \in \Sigma$ and $x \in \Sigma^\star$, then $|w| = 1 + |x|$. For each σ in Σ and each w in $\Sigma^\star$, let $\#_\sigma(w)$ be the number of times that the symbol σ occurs in the word w.

An (ordinary or crisp formal) *language* over Σ is a subset of $\Sigma^\star$. A language L is λ-*free* if it does not contain λ, i.e., if $L \subseteq \Sigma^+$.

Example 2.1. Let Σ be the alphabet $\{a, b\}$. Then λ, aab, and $babb$ are words over Σ of length 0, 3 and 4, respectively. We have $\#_a(\lambda) = 0$, $\#_a(aab) = 2$ and $\#_a(babb) = 1$.

The set $L_0 = \{w \mid\ w \in \{a, b\}^+,\ \#_a(w) = \#_b(w)\}$ is a λ-free language over Σ. Note that for each w in L_0, $|w|$ is even. □

Fuzzy languages have been originally introduced in [20] in which the characteristic function $\mu_{L_0} : \Sigma^\star \to \{0, 1\}$ of a language L_0 over Σ has been generalized to the (degree of) membership function $\mu_{L_0} : \Sigma^\star \to [0, 1]$. In [6,7] we replaced the interval [0,1] by a more general lattice-ordered structure in order to model errors in grammatical and parallel rewriting; cf. also [18,23]. Many definitions and examples in this section and the next one are quoted from [6,7].

Definition 2.2. An algebraic structure $\mathcal{L}$ or $(\mathcal{L}, \wedge, \vee, 0, 1, \star)$ is a *type-00 lattice* if it satisfies the following conditions.

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

$$a \star \bigvee_i b_i = \bigvee_i (a \star b_i), \qquad (\bigvee_i a_i) \star b = \bigvee_i (a_i \star b),$$
$$0 \wedge a = 0 \star a = a \star 0 = 0, \qquad 1 \wedge a = 1 \star a = a \star 1 = a.$$

A *type-01 lattice* is a type-00 lattice in which the operations $\star$ and $\wedge$ coincide; 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, 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*. □

The above definition of type 00-lattice is a slight modification of a structure originally introduced in [13]; cf. also [18,23].

Example 2.3. (1) The structure $([0, 1] \times [0, 1], \wedge, \vee, (0, 0), (1, 1), \star)$, with operations $\wedge$, $\vee$ and $\star$ defined by

$(x_1, y_1) \vee (x_2, y_2) = (\max\{x_1, x_2\}, \max\{y_1, y_2\})$,

$(x_1, y_1) \wedge (x_2, y_2) = (\min\{x_1, x_2\}, \min\{y_1, y_2\})$ and

$(x_1, y_1) \star (x_2, y_2) = (x_1x_2, y_1y_2)$

for all x_1, x_2, y_1 and y_2 in $[0, 1]$, is a type-00 lattice.

(2) Then $([0, 1] \times [0, 1], \wedge, \vee, (0, 0), (1, 1), \star)$, where the operations $\wedge$ and $\vee$ are

defined as in (1) and $(x_1, y_1) \star (x_2, y_2) = (\min\{x_1, x_2\}, \min\{y_1, y_2\})$ for all x_1, x_2, y_1 and y_2 in $[0, 1]$, is a type-01 lattice.

(3) The structure $([0, 1], \min, \max, 0, 1, \star)$ 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) A type-11 lattice is obtained by taking $\star$ equal to min in (3). □

The following elementary fact is very useful in the sequel.

Lemma 2.4. [6,7] *For each type-00 lattice $\mathcal{L}$, $a \star b \leq a \wedge b$ holds for all a and b in $\mathcal{L}$. Consequently, $a \star b \leq a$ also holds for all a and b in $\mathcal{L}$.* □

3 Fuzzy Languages and Operations on Fuzzy Languages

As mentioned above the notion of fuzzy language (Definition 3.1 below) will be based on the lattice-ordered structures of Definition 2.2 rather than the real closed interval $[0, 1]$.

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

When $\mathcal{L}$ is clear from the context, we use "fuzzy language" instead of "$\mathcal{L}$-fuzzy language". We will often write $\mu(x; L)$ rather than $\mu_L(x)$ in order to reduce the number of subscript levels.

Each ordinary (non-fuzzy) language L coincides with its crisp part $c(L)$. Therefore an ordinary language will also be called a *crisp* language. □

In dealing with fuzzy languages (Σ, μ_L) the degree of membership function μ_L is actually the principal concept, whereas its support $s(L)$, its crisp part $c(L)$ and many other crisp languages like $L^{\geq a} = \{w \in \Sigma^\star \mid \mu(w; L) \geq a\}$, $L^{<a} = \{w \in \Sigma^\star \mid \mu(w; L) < a\}$ and $L^{a \leq ; \leq b} = \{w \in \Sigma^\star \mid a \leq \mu(w; L) \leq b\}$ are derived notions (a and b are elements in $\mathcal{L}$).

Example 3.2. (1) Let Σ be the alphabet $\{a, b\}$, let $\mathcal{L}$ be the type 11-lattice of Example 2.3(4), and consider the $\mathcal{L}$-fuzzy language L_1 over Σ, where μ_{L_1} is defined for $w \in \{a, b\}^\star$ by

- $\mu(w; L_1) = 1$ if and only if $\#_a(w) = \#_b(w)$ and $w \neq \lambda$,
- $\mu(w; L_1) = 0.9$ if and only if $\#_b(w) \geq \#_a(w) + 2$ and $|w|$ is even,
- $\mu(w; L_1) = 0.1$ if and only if $\#_a(w) \geq \#_b(w) + 2$ and $|w|$ is even,
- $\mu(w; L_1) = 0$ if and only if either $w = \lambda$ or $|w|$ is odd.

Then $c(L_1) = L_0$ where L_0 is the crisp language of Example 2.1.

(2) Let $\mathcal{L}$ be the type-00 lattice of Example 2.3(1). Consider the $\mathcal{L}$-fuzzy language L_2 over $\Sigma = \{a, b\}$ defined by

$\mu(a^m b^n; L_2) = \left(\frac{m}{\max\{1,m,n\}}, \frac{n}{\max\{1,m,n\}}\right)$,   if $m, n \geq 0$.

In defining the degree of membership function is such a concrete case, we always tacitly assume that $\mu(x; L_2)$ equals the zero element of $\mathcal{L}$ in all other, unmentioned cases for x in $\Sigma^\star$. Consequently, we have, e.g., $\mu(b^2a^4; L_2) = \mu(a^3b^2a^5; L_2) = \mu(ab^4a^3b^2; L_2) = (0,0)$.

Then the crisp part of L_2 equals $c(L_2) = \{a^m b^m \mid m \geq 1\}$: for each x in $c(L_2)$, we have $\mu(x; L_2) = (1,1)$. Note that for each $m \geq 1$, $\mu(a^m; L_2) = (1,0)$ and $\mu(b^m; L_2) = (0,1)$, whereas for the empty word λ, we have $\mu(\lambda; L_2) = (0,0)$.

(3) Now we take for $\mathcal{L}$ the type-10 lattice of Example 2.3(3). Let L_3 be the fuzzy language over $\{a,b\}$ defined by

$\mu(w; L_3) =$ **if** $|w| = 2^k$ for some $k \geq 0$ **then** $2^{-\#_b(w)}$ **else** 0.

Then the fuzzy language L_3 satisfies $s(L_3) = \{w \mid w \in \{a,b\}^+,\ |w| = 2^k$ for some $k \geq 0\}$, and $c(L_3) = \{a^{2^k} \mid k \geq 0\}$. □

Remark. Since in many of our examples the function μ has as (a part of) its codomain the closed interval $[0,1]$, each real number from this interval may occur as the value for some string x. However, using non-computable reals as a cut point or as a threshold in specifying a fuzzy language may give rise to problems of an undecidable nature, i.e., to languages that are not recursively enumerable [10]. In the sequel we avoid this problem by restricting ourselves to the computable, or even to the rational elements of $[0,1]$ only. □

For an account on the impact of computability constraints in fuzzy formal languages we refer the reader to [10].

Note that two fuzzy languages $L_1 = (\Sigma_1, \mu_{L_1})$ and $L_2 = (\Sigma_2, \mu_{L_2})$ are *equal*, denoted by $L_1 = L_2$, if $\Sigma_1 = \Sigma_2$ and $\mu_{L_1} = \mu_{L_2}$, i.e., if for all $x \in (\Sigma_1 \cup \Sigma_2)^\star$, $\mu(x; L_1) = \mu(x; L_2)$. Of course, equality ($L_1 = L_2$) implies equality of supports ($s(L_1) = s(L_2)$) and of crisp parts ($c(L_1) = c(L_2)$), but not vice versa. See also Example 4.5 below.

Starting from simple fuzzy languages we can define more complicated ones by applying operations on fuzzy languages. First, we consider the operations union, intersection and concatenation for fuzzy languages; they have been defined originally in [20] for the type-11 lattice $[0,1]$; cf. Example 2.3(4) and [23]. In [5] we remarked that a generalization to the type-10 lattice of Example 2.3(3) is possible. However, it is easy to define these operations for arbitrary type-00 lattices; cf. [6,7] from which we quote the following definitions.

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 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 for all x in $(\Sigma_1 \cup \Sigma_2)^\star$ by

$\mu(x; L_1 \cup L_2) = \mu(x; L_1) \vee \mu(x; L_2)$,

$\mu(x; L_1 \cap L_2) = \mu(x; L_1) \wedge \mu(x; L_2)$, and

$\mu(x; L_1L_2) = \bigvee\{\mu(y; L_1) \star \mu(z; L_2) \mid x = yz\}$.

Once we have defined union and concatenation it is straightforward to define the operations of *Kleene* + and *Kleene* $\star$ for a fuzzy language L; viz. by

$L^+ = L \cup LL \cup LLL \cup \cdots = \bigcup\{L^i \mid i \geq 1\}$, and

$L^\star = \{\lambda\} \cup L \cup LL \cup LLL \cup \cdots = \bigcup\{L^i \mid i \geq 0\}$, respectively,

where $L^0 = \{\lambda\}$, and $L^{n+1} = L^nL$ with $n \geq 0$. Clearly, we have for $n \geq 0$,

$\mu(x; L^n) = \bigvee\{\mu(x_1; L) \star \mu(x_2; L) \star \cdots \star \mu(x_n; L) \mid x_1x_2\cdots x_n = x\}$, and

$\mu(x; L^\star) = \bigvee\{\mu(x_1; L) \star \mu(x_2; L) \star \cdots \star \mu(x_n; L) \mid n \geq 0,\ x_1x_2\cdots x_n = x\}$.

Then $\mu(\lambda; L^0) = 1$, since $x_1x_2\cdots x_n = \lambda$ and $a_1 \star a_2 \star \cdots \star a_n = 1$ in case $n = 0$ $(a_1, a_2, \ldots, a_n \in \mathcal{L})$, and so $\mu(\lambda; L^\star) = 1$. Hence $L^\star = L^+ \cup \{\lambda\}$ where the latter set in this union is crisp.

Remark. To avoid technical problems we require the following convention: if a fuzzy language L contains λ, then $\mu(\lambda; L) = 1$. So for each fuzzy language L, we have $\mu(\lambda; L) \in \{0, 1\}$. □

Example 3.3. L_1 from Example 3.2(1) satisfies the equality $L_1^+ = L_1$, but L_1 is a proper subset of $L_1^\star$. □

Apart from these simple operations on fuzzy languages we need some other well-known ones, like homomorphisms and substitutions. They can be extended from crisp to fuzzy languages by means of the concept of fuzzy function; cf. [6] and [7] for the original definitions.

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 a fuzzy subset of $X \times Z$, defined by

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

A *fuzzy function* $f : X \to Y$ is a fuzzy relation $f \subseteq X \times Y$, satisfying the condition 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 usually write the composition of two functions $f : X \to Y$ and $g : Y \to Z$ as $g{\circ}f : X \to Z$ rather than as $f{\circ}g$.

Let $\mathbb{F}(X)$ denote the power set of the fuzzy set X, i.e., the collection of all fuzzy subsets of the fuzzy set X:

$$\mathbb{F}(X) = \{Y \mid \forall x \in X, \mu(x, Y) \leq \mu(x; X);\ \forall x \notin X : \mu(x; Y) = 0\}.$$

In the sequel we need functions of type $f : V^\star \to \mathbb{F}(V^\star)$, where V is an alphabet, that will be extended to a function of type $f : \mathbb{F}(V^\star) \to \mathbb{F}(V^\star)$ by $f(L) = \bigcup\{f(x) \mid x \in L\}$ and for each language L over V,

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

Consequently, by (1) and (2) iterating a single fuzzy function f, yielding functions like $f{\circ}f$, $f{\circ}f{\circ}f$, and so on, is now defined. Each of these functions $f^{(k)}$

is of type $f^{(k)} : \mathbb{F}(V^\star) \to \mathbb{F}(V^\star)$. Of course, we can iterated a finite set of such functions $\{f_1, \ldots, f_n\}$ in the very same way.

4 Fuzzy Context-Free Grammars

The notion of fuzzy context-free grammar has originally been introduced in [20]. However, in Definition 4.3 below, we define fuzzy context-free grammars in a slightly different way, but it is easy to show that both definitions are equivalent. Definition 4.3 uses operations like concatenation and intersection of fuzzy languages and is a better starting point for introducing the generalized fuzzy context-free grammars of Section 5. First, we will reconsider (ordinary or crisp) context-free grammars, then we will turn to their fuzzy counterparts.

Recall that a *context-free grammar* $G = (V, \Sigma, P, S)$ consists of an alphabet V, a terminal alphabet Σ ($\Sigma \subseteq V$), a finite set P of productions or rules ($P \subseteq N \times V^\star$, where $N = V - \Sigma$ is the set of nonterminal symbols of G), and an initial symbol S ($S \in N$). Usually, a production (A, ω) is written as $A \to \omega$, and all rules $A \to \omega_1$, $A \to \omega_2$, ..., $A \to \omega_n$ with the same left-hand side A are collected in a single expression of the form $A \;\to\; \omega_1 \mid \omega_2 \mid \ldots \mid \omega_n$.

A context-free grammar $G = (V, \Sigma, P, S)$ gives rise to a *derivation* relation $\Rightarrow$ and a *language* $L(G)$ generated by G. Formally, $\varphi_1 \Rightarrow \varphi_2$ holds for words $\varphi_1, \varphi_2 \in V^\star$ if and only if there exist words $u, v \in V^\star$ and a rule $A \to \omega$ in P such that $\varphi_1 = uAv$ and $\varphi_2 = u\omega v$. Then $L(G)$ is defined by $L(G) = \{w \in \Sigma^\star \mid S \Rightarrow^\star w\}$, where $\Rightarrow^\star$ is the reflexive and transitive closure of $\Rightarrow$.

Example 4.1. Let $\Sigma = \{a, b\}$, $N = \{S, A, B\}$ and $V = N \cup \Sigma$. Consider the context-free grammars $G_1 = (V, \Sigma, P_1, S)$ and $G_2 = (V, \Sigma, P_2, S)$ where P_1 and P_2 are given by

P_1:			P_2:		
S	$\to$	$AB \mid BA$	S	$\to$	$aSB \mid aBS \mid bSA \mid bAS \mid aB \mid bA$
A	$\to$	$AS \mid SA \mid a$	A	$\to$	$aS \mid a$
B	$\to$	$BS \mid SB \mid b$	B	$\to$	$bS \mid b.$

Then $S \Rightarrow BA \Rightarrow BSA \Rightarrow BSa \Rightarrow BABa \Rightarrow bABa \Rightarrow bAba \Rightarrow baba$ and $S \Rightarrow bSA \Rightarrow bSa \Rightarrow bbAa \Rightarrow bbaa$ are derivations according to G_1 and G_2, respectively. It is straightforward to prove that G_1 and G_2 both generate the crisp language L_0 of Example 2.1: $L(G_1) = L(G_2) = L_0$. □

Our first step in defining fuzzy context-free grammars consists of redefining crisp context-free grammars slightly. Viz. we define, given $G = (V, \Sigma, P, S)$, for each symbol α in V,

$$P(\alpha) \;=\; \{\omega \mid \alpha \to \omega \in P\} \cup \{\alpha\},$$

i.e., $P(\alpha)$ is the set consisting of α together with all right-hand sides of those rules in P with left-hand side equal to α. Thus for each α, $P(\alpha)$ is a finite language over V that contains α. And $P(\alpha) = \{\alpha\}$ whenever α belongs to Σ.

The next step is that we consider P as a mapping from V to the family of finite languages over V. This mapping can be extended to words over V by

- $P(\lambda) = \{\lambda\}$, and
- $P(\alpha_1 \cdots \alpha_n) = P(\alpha_1) \cdots P(\alpha_n)$ where $\alpha_i \in V$ $(1 \le i \le n)$,

and to languages L over V by

- $P(L) = \bigcup\{P(x) \mid x \in L\}$.

Such a mapping P is called a *nested finite substitution* over V [14,22,2,3], since P is a finite substitution (i.e., each $P(\alpha)$ is a finite language) that is nested (i.e., $\alpha \in P(\alpha)$ for each α in V). And it can be *iterated*:

- $P^0(x) = \{x\}$,
- $P^{i+1}(x) = P(P^i(x))$, and
- $P^\star(x) = \bigcup\{P^i(x) \mid i \ge 0\}$.

Then it is straightforward to prove that for each context-free grammar $G = (V, \Sigma, P, S)$, we have $L(G) = P^\star(S) \cap \Sigma^\star$. Now $L(G)$ is defined in terms of set-theoretical operations only rather than using the concept of derivation. Moreover, these operations on crisp sets can be easily replaced by their fuzzy counterparts introduced in Section 3; cf. Definition 4.3 below.

Example 4.2. Viewing P_1 and P_2 of Example 4.1 as nested finite substitutions over the alphabet $\{S, A, B, a, b\}$ yields

$P_1(S) = \{S, AB, BA\}$	$P_2(S) = \{S, aSB, aBS, bSA, bAS, aB, bA\}$
$P_1(A) = \{A, AS, SA, a\}$	$P_2(A) = \{A, aS, a\}$
$P_1(B) = \{B, BS, SB, b\}$	$P_2(B) = \{B, bS, b\}$
$P_1(a) = \{a\}$	$P_2(a) = \{a\}$
$P_1(b) = \{b\}$	$P_2(b) = \{b\}$

□

The last step is to replace the crisp finite sets $P(\alpha)$ $(\alpha \in V)$ in the definition of context-free grammar by fuzzy finite sets.

Definition 4.3. A *fuzzy context-free grammar* G is a 4-tuple (V, Σ, P, S) where V, Σ and S are as usual, and for each α in V, $P(\alpha)$ is a fuzzy subset of $V^\star$ satisfying

(1) $\mu(\alpha; P(\alpha)) = 1$, i.e., P is *nested*,

(2) the support of $P(\alpha)$, i.e. $s(P(\alpha)) = \{\omega \mid \mu(\omega; P(\alpha)) > 0\}$, is finite, and

(3) the support of $P(\alpha)$ equals $\{\alpha\}$ in case α belongs to Σ: $s(P(\alpha)) = \{\alpha\}$.

The fuzzy context-free *language* generated by G is the fuzzy subset $L(G)$ of $\Sigma^\star$ defined by $L(G) = P^\star(S) \cap \Sigma^\star$. A fuzzy language L_0 is called a *fuzzy context-free language* if there exists a fuzzy context-free grammar G with $L(G) = L_0$. The family of all fuzzy context-free languages is denoted by CF_f. □

In the expression "$P^\star(S) \cap \Sigma^\star$" all operations involved are operations on fuzzy sets (intersection as well as union, concatenation and composition of fuzzy functions via $P^\star$; cf. Section 3), although $\Sigma^\star$ happens to be a crisp set.

Note that, if we replace in a fuzzy context-free grammar each fuzzy set $P(\alpha)$ by a crisp finite language over V, then we reobtain an ordinary context-free grammar. The family of crisp context-free languages will be denoted by CF. Then we have $\mathrm{CF} \subseteq \{c(L) \mid L \in \mathrm{CF}_f\}$.

When $\mathcal{L}$ equals the type-11 lattice of Example 2.3(4) it is a routine matter to show that Definition 4.3 is equivalent to the one of [20]. Then $L(G)$ can also be defined in terms of derivations consisting of rules of G that are applied consecutively [20]. And a string x over Σ belongs to the fuzzy language $L(G)$ if and only if there exist strings $\omega_0, \omega_1, \ldots, \omega_n$ over V such that $S = \omega_0 \Rightarrow \omega_1 \Rightarrow \omega_2 \Rightarrow \cdots \Rightarrow \omega_n = x$. If $A_i \to \psi_i$ $(0 \leq i < n)$ are the respective productions used in this derivation, then the degree of membership of x in $L(G)$ is

$$\mu(x; L(G)) = \max\{\min\{\mu(\psi_i; P(A_i)) \mid 0 \leq i < n\} \mid S = \omega_0 \Rightarrow^\star \omega_n = x\ \}.$$

When such a derivation is viewed as a chain link of rule applications, its total "strength" equals the strength of its weakest link; hence the min-operation. And $\mu(x; L(G))$ is the strength of the strongest derivation chain from S to x: the maximum is taken over all possible derivations of x from S [20].

Henceforth, we use $X = \{x_1/_{m_1}, \ldots, x_n/_{m_n}\}$ as a concise representation of the finite fuzzy set $X = \{x_1, \ldots, x_n\}$ with $\mu(x_i; X) = m_i$ $(1 \leq i \leq n)$.

Example 4.4. Let $\mathcal{L}$ be the type-11 lattice of Example 2.3(4) and $G_3 = (V, \Sigma, P_3, S)$ the $\mathcal{L}$-fuzzy context-free grammar with $N = V - \Sigma = \{S, A, B\}$, $\Sigma = \{a, b\}$, and P_3 is defined by

$$P_3(S) = \{S/_1, AB/_1, BA/_1, AA/_{0.1}, BB/_{0.9}\}, \qquad P_3(a) = \{a/_1\},$$
$$P_3(A) = \{A/_1, AS/_1, SA/_1, a/_1\}, \qquad P_3(b) = \{b/_1\},$$
$$P_3(B) = \{B/_1, BS/_1, SB/_1, b/_1\}.$$

The crisp language $c(L(G_3))$ is generated by the (ordinary) context-free grammar G_1 of Example 4.1; cf. also Example 4.2.

So G_3 describes the set of all nonempty even length strings over $\{a, b\}$ with preferably as many a's as b's (degree of membership equal to 1). Occasionally, some a's in these nonempty even length strings may be changed into b's or vice versa, due to grammatical errors modeled by the rules $S \to BB$ and $S \to AA$, respectively. The former error happens to be a quite less severe incident than the latter (degrees of membership 0.9 and 0.1, respectively). It is easy to show that $L(G_3) = L_1$, where L_1 is the language from Example 3.2(1). □

Modeling grammatical errors as in Example 4.4 has a serious shortcoming: making the same error twice (or many more times) does not decrease the degree of membership as one would expect intuitively; cf. [4,5]. Actually, a fixed finite set of rationals —viz. $\{0,\ 0.1,\ 0.9,\ 1\}$— serves as codomain of the function $\mu_{L(G_3)}$; cf. also Example 3.2(1). Obviously, the operations min and max applied to this set do not yield any new, different values in this codomain. Augmenting $\mathcal{L}$ with an operation $\star$ different from min enables us to model grammatical errors more adequately; cf. Lemma 2.4.

Example 4.5. Consider the $\mathcal{L}$-fuzzy context-free grammar $G_4 = (V, \Sigma, P_4, S)$ which is equal to G_3 of Example 4.4 except that $\mathcal{L}$ is now the type-10 lattice of Example 2.3(3) rather than the type-11 lattice of Example 2.3(4). Then we have for w in $\{a, b\}^\star$,

- $\mu(w; L(G_4)) = 1$ iff $\#_a(w) = \#_b(w)$ and $w \neq \lambda$,
- $\mu(w; L(G_4)) = (\frac{9}{10})^{(\#_b(w)-\#_a(w))/2}$ iff $\#_b(w) \geq \#_a(w) + 2$ and $|w|$ is even,
- $\mu(w; L(G_4)) = (\frac{1}{10})^{(\#_a(w)-\#_b(w))/2}$ iff $\#_a(w) \geq \#_b(w) + 2$ and $|w|$ is even,
- $\mu(w; L(G_4)) = 0$ iff either $w = \lambda$ or $|w|$ is odd.

So the fuzzy languages $(\Sigma, \mu_{L(G_4)})$ and (Σ, μ_{L_1}) of Example 3.2(1) are different, since $\mu_{L(G_4)}$ and μ_{L_1} differ: *not* for all w in $\Sigma^\star$, $\mu(w; L(G_4)) = \mu(w; L_1)$ or, equivalently, $L(G_4) \neq L_1$. But their crisp parts and supports still coincide: $c(L(G_4)) = c(L_1)$ and $s(L(G_4)) = s(L_1)$. □

Note that the codomain of $\mu_{L(G_4)}$ in Example 4.5 is a countably infinite set of rationals. And the accumulation of grammatical errors results in strings, still belonging to the support of $\mu_{L(G_4)}$, of which the degree of membership strictly decreases as the number of grammatical errors increases; cf. Lemma 2.4.

5 Fuzzy Context-Free K-grammars

In this section we continue to address the question how "tiny mistakes" and "capital blunders" can be described by (generalized) fuzzy context-free grammars. Our ultimate main goal is to determine the expressive power of these generalized fuzzy context-free grammars; cf. Theorem 7.1 below.

To be more concrete, let us return to Example 4.4. The principal aim of the fuzzy context-free grammar G_3 is to generate the (crisp) language

$L_1 = L(G_1) = \{w \mid w \in \{a, b\}^+,\ \#_a(w) = \#_b(w)\} = c(L(G_3))$.

However, applying the rule $S \to BB$ instead of either $S \to AB$ or $S \to BA$ one or more times during a derivation, results in a terminal string w that satisfies: $\#_b(w) \geq \#_a(w) + 2$, $|w|$ is even, and $\mu(w; L(G_3)) = 0.9$. So such terminal strings w may be considered as tiny mistakes. On the other hand, using the rule $S \to AA$ instead of either $S \to AB$ or $S \to BA$ one or more times in a derivation, yields a terminal string w that satisfies: $\#_a(w) \geq \#_b(w) + 2$, $|w|$ is even, and $\mu(w; L(G_3)) = 0.1$. Terminal strings w of this type may be viewed as capital blunders, since they "hardly belong" to the fuzzy language $L(G_3)$.

In Example 4.5 we may encounter the situation that due to the accumulation of errors in a long sequence of tiny mistakes we end up with a terminal string that looks like a capital blunder. And in both Examples 4.4 and 4.5

using an erroneous rule $S \to BB$ can be compensated by the application of an erroneous rule $S \to AA$ (and vice versa) resulting in a correct string x, i.e., $\mu(x; L(G_3)) = \mu(x; L(G_4)) = 1$, as there also exists a "completely correct" derivation for x that determines its degree of membership.

Note that P_3 is obtained from P_1 by adding the rules $S \to AA$ and $S \to BB$ with degree of membership 0.1 and 0.9, respectively: so $P_3(S) = P_1(S) \cup E_1$ with $E_1 = \{AA/_{0.1}, BB/_{0.9}\}$. But the union of two finite fuzzy sets is a finite fuzzy set; so (2) of Definition 4.3 is not violated and we remain within the framework of fuzzy context-free grammars.

Now the question arises: to what extend can we proceed in this way? Or: what are the limits of the fuzzy context-free framework in describing errors? In Examples 4.4 and 4.5 we modeled the situation of two ways to apply a rule erroneously. But in principal there are infinitely many ways to make an error, although substituting a very long word instead of a short one is rather unlikely. So what happens, for instance, when we add an infinite fuzzy set E_1 to $P_1(S)$ instead of a finite one? Or, equivalently, when we replace the finite fuzzy sets $P(\alpha)$ (for each α in V) by infinite ones satisfying $\mu(\alpha; P(\alpha)) = 1$? Unfortunately, this will not work, since then the languages $L(G'_3)$ and $L(G'_4)$ generated by the resulting respective grammars G'_3 and G'_4 might not even be recursively enumerable [10]. Thus we have to restrain the languages $P(\alpha)$ in some, preferably natural, way. The method we use here, originates from [22]: we assume that a family K of fuzzy languages is given in advance, from which we are allowed to take whatever languages we think to be appropriate. Then replacing the finite languages $P(\alpha)$ over V by members from the family K, yields the concept of fuzzy context-free K-grammar (Definition 5.3). The family K plays the rôle of parameter, and when we take K equal to the constant value FIN_f, the family of finite fuzzy languages, we reobtain the ordinary fuzzy context-free grammars. In this approach we need the notions of family of fuzzy languages (Definition 5.1) and of fuzzy K-substitution (Definition 5.2).

Definition 5.1. Let Σ_ω be a countably infinite set of symbols. As usual a *family of languages* over Σ_ω is a set of pairs (Σ_L, L) where L is a crisp language over Σ_L and Σ_L is a finite subset of Σ_ω. The set Σ_L is assumed to be the minimal alphabet of L.

Similarly, a *family of fuzzy languages* K is a set of fuzzy languages (Σ_L, μ_L) such that each Σ_L is a finite subset of Σ_ω. As usual, we assume that for each fuzzy language (Σ_L, μ_L) in K, Σ_L is minimal with respect to μ_L, 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 $\mu_L(w) > 0$ or, equivalently, for which $w \in s(L)$ holds.

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

The *crisp part* $c(K)$ of a family K is defined by $c(K) = \{c(L) \mid L \in K\}$. □

Henceforth we assume that each family K of (fuzzy) languages is normalized and closed under isomorphism; thus for each language L in K over some alphabet Σ and for each bijective mapping $i : \Sigma \to \Sigma_1$ —extended to words and to languages in the usual way— the language $i(L)$ belongs to K.

Concrete examples of simple, normalized families of fuzzy languages, which we will need in the sequel, are the family FIN_f of finite fuzzy languages

$\mathrm{FIN}_f = \{(\Sigma_L, \mu_L) \mid \Sigma_L \subset \Sigma_\omega,\ s(L) \text{ is finite}\}$,

the family ONE_f of singleton fuzzy languages

$\mathrm{ONE}_f = \{(\Sigma_L, \mu_L) \mid \Sigma_L \subset \Sigma_\omega,\ s(L) \text{ is a singleton}\}$,

the family ALPHA_f of fuzzy alphabets

$\mathrm{ALPHA}_f = \{(\Sigma_L, \mu_L) \mid \Sigma_L \subset \Sigma_\omega,\ s(L) = \Sigma_L\}$,

and the family SYMBOL_f of singleton fuzzy alphabets

$\mathrm{SYMBOL}_f = \{(\Sigma_L, \mu_L) \mid \Sigma_L \subset \Sigma_\omega,\ s(L) = \Sigma_L,\ s(L) \text{ is a singleton}\}$.

The crisp counterparts of these language families are denoted by FIN, ONE, ALPHA, and SYMBOL, respectively. Clearly, the equality $c(\mathrm{FIN}_f) = \mathrm{FIN}$ holds, as well as similar statements for these other simple families.

The concept of fuzzy substitution is defined in a way very similar to the notion of substitution for crisp languages; cf. [6–8].

Definition 5.2. Let K be a family of fuzzy languages and let V be an alphabet. A mapping $\tau : V \to K$ is called a *fuzzy K-substitution on V*; it is extended to words over V by $\tau(\lambda) = \{\lambda/_1\}$, and $\tau(\alpha_1 \ldots \alpha_n) = \tau(\alpha_1) \cdots \tau(\alpha_n)$ where $\alpha_i \in V$ $(1 \leq i \leq n)$, and to languages L over V by $\tau(L) = \bigcup\{\tau(w) \mid w \in L\}$. If for each $\alpha \in V$, $s(\tau(\alpha)) \subseteq V^\star$, then $\tau : V \to K$ is called a *fuzzy K-substitution over V*. A fuzzy K-substitution τ over V is called *nested* if $\mu(\alpha; \tau(\alpha)) = 1$ for each α in V. □

When we take K equal to a family of crisp languages, we obtain the well-known definition of substitution. So a ONE-substitution is a homomorphism, and one-to-one SYMBOL-substitution is an isomorphism ("renaming of symbols"). And a fuzzy ONE_f-substitution will be called a fuzzy homomorphism.

Definition 5.3. Let K be a family of fuzzy languages. A *fuzzy context-free K-grammar* $G = (V, \Sigma, U, S)$ consists of an alphabet V, a terminal alphabet Σ $(\Sigma \subseteq V)$, a start symbol S $(S \in V)$, and a finite set U of nested fuzzy K-substitutions over V. So each element τ of U is a mapping $\tau : V \to K$ satisfying: for each symbol α in V, $\tau(\alpha)$ is a fuzzy language over the alphabet V from the family K with $\mu(\alpha; \tau(\alpha)) = 1$.

The fuzzy *language* generated by G is the fuzzy set $L(G)$ defined by

$L(G) = U^\star(S) \cap \Sigma^\star = \bigcup\{\tau_n \circ \cdots \circ \tau_1(S) \mid \tau_i \in U,\ 1 \leq i \leq n,\ n \geq 0\} \cap \Sigma^\star$.

Two fuzzy context-free K-grammars G_1 and G_2 are *equivalent* if $L(G_1) = L(G_2)$.

The family of fuzzy languages generated by fuzzy context-free K-grammars is denoted by $A_f(K)$. The family of fuzzy languages generated by fuzzy context-free K-grammars that possess at most m ($m \geq 1$) elements in U is denoted by $A_{f,m}(K)$. Consequently, $A_f(K) = \bigcup\{A_{f,m}(K) \mid m \geq 1\}$. □

Note that the families of crisp languages corresponding to the families $A_f(K)$ and $A_{f,m}(K)$ are $c(A_f(K)) = \{c(L) \mid L \in A_f(K)\}$ and $c(A_{f,m}(K)) = \{c(L) \mid L \in A_{f,m}(K)\}$, respectively.

Replacing K in Definition 5.3 by a family of crisp languages results in the definition of *context-free K-grammar* [22,2]; the corresponding family of languages is denoted by $A(K)$. Obviously, if K is a family of crisp languages, then $A(K) = A_f(K)$. In case K is a family of $\mathcal{L}$-fuzzy languages, where $\mathcal{L}$ is a type-00 lattice, then we have $A(c(K)) \subseteq c(A_f(K)) \subseteq A_f(K)$, which implies $\mathrm{CF} = A(\mathrm{FIN}) \subseteq c(A_f(\mathrm{FIN}_f)) = c(\mathrm{CF}_f) \subseteq \mathrm{CF}_f$; cf. Corollary 7.2 below.

If $\mathcal{L}$ is linearly ordered, i.e. if $\mathcal{L}$ is a type-10 lattice, we have the equality: $A(c(K)) = c(A_f(K))$. On the other hand, if $K = \mathrm{FIN}_f$ and $\mathcal{L}$ equals the four element distributive lattice that is not a chain —i.e., $\mathcal{L} = \{0, \xi, \eta, 1\}$ with $0 < \xi < 1$, $0 < \eta < 1$ whereas ξ and η are incomparable— we can show that $\mathrm{CF} = A(\mathrm{FIN}) \subset c(A_f(\mathrm{FIN}_f)) = c(\mathrm{CF}_f) \subseteq \mathrm{CF}_f$; cf. [8].

Example 5.4. Let $\mathcal{L}$ be the type-11 lattice of Example 2.3(4) and $G_5 = (V, \Sigma, \{\tau_5\}, S)$ the $\mathcal{L}$-fuzzy context-free CF_f-grammar with $N = V - \Sigma = \{S, A, B\}$, $\Sigma = \{a, b\}$, and

$$\tau_5(S) = P_3(S) \cup L_2 \cup L_3 \cup L_4$$
$$\tau_5(\alpha) = P_3(\alpha) \qquad \alpha \neq S$$

where P_3 is as in Example 4.4, $s(L_2) = \{aA^n bB^n \mid n \geq 1\}$, $s(L_3) = \{aA^{2n-1} \mid n \geq 2\}$, and $s(L_4) = \{B^{2n} \mid n \geq 3\}$. The degrees of membership are as in Example 4.4 together with

$\mu(aA^m bB^m; L_2) = 1 \qquad (m \geq 1)$,

$\mu(aA^m; L_3) =$ **if** m is odd and $m \geq 2$ **then** 0.1 **else** 0,

$\mu(B^m; L_4) =$ **if** m is even and $m \geq 6$ **then** 0.9 **else** 0.

Since $L(G_5) = L(G_3)$, G_5 and G_3 (Example 4.4) are equivalent. □

Example 5.5. Let $\mathcal{L}$ be the type-10 lattice of Example 2.3(3) and $G_6 = (V, \Sigma, \{\tau_6\}, S)$ the $\mathcal{L}$-fuzzy context-free CF_f-grammar with $N = V - \Sigma = \{S, A, B\}$, $\Sigma = \{a, b\}$, and

$$\tau_6(S) = P_4(S) \cup L_2 \cup L_3 \cup L_4$$
$$\tau_6(\alpha) = P_4(\alpha) \qquad \alpha \neq S$$

where P_4 is as in Example 4.5, and $s(L_2)$, $s(L_3)$ and $s(L_4)$ are as in Example 5.4. Most degrees of membership have been given in Example 4.5, except for

$\mu(aA^m bB^m; L_2) = 1 \qquad (m \geq 1)$,

$\mu(aA^m; L_3) =$ **if** m is odd and $m \geq 2$ **then** $(\frac{1}{10})^{(m+1)/2}$ **else** 0,

$\mu(B^m; L_4) =$ **if** m is even and $m \geq 6$ **then** $(\frac{9}{10})^{m/2}$ **else** 0.

Then G_6 and G_4 of Example 4.5 are equivalent: $L(G_6) = L(G_4)$. □

6 Elementary Properties

Comparing Definitions 4.3 and 5.3 shows that we removed the requirements (2) and (3) in 4.3 to obtain 5.3, and we use a finite set of nested fuzzy K-substitutions rather than a single fuzzy finite substitution. Now (3) is just a minor point as we will see in Lemma 6.1. Using a finite number rather than a single substitution is neither a proper extension (Lemma 6.2). So removing (2) in Definition 4.3 is the main point: we replace finite fuzzy languages in Definition 4.3 by (not necessarily finite) fuzzy languages from a given family K. This latter aspect is the main feature of fuzzy context-free K-grammars.

Now we turn to a few lemmas needed to establish the main result of this paper (Theorem 7.1).

Lemma 6.1. *Let K be a family of fuzzy languages that is closed under union with languages from* SYMBOL. *If $K \supseteq$* SYMBOL, *then for each fuzzy context-free K-grammar $G_1 = (V_1, \Sigma, U_1, S)$, there exists an equivalent fuzzy context-free K-grammar $G_2 = (V_2, \Sigma, U_2, S)$ such that for each τ in U_2, $\tau(\alpha) = \{\alpha/_1\}$ in case α belongs to Σ.*

Proof. We introduce for each a in Σ a new nonterminal symbol A_a with for each τ in U_1, $\tau(A_a) = \{A_a/_1, a/_1\}$. Next we replace each occurrence of a by A_a by means of the isomorphism $i(a) = A_a$. Thus the language $\tau(\alpha)$ from K becomes the language $i(\tau(\alpha))$ for each $\tau \in U_1$ and each $\alpha \in V_1$. This language $i(\tau(\alpha))$ is in K too, since we assumed that all language families are closed under isomorphism. Consequently, $\mu(\omega; \tau(\alpha)) = \mu(i(\omega); i(\tau(\alpha)))$ for each $\omega \in V_1^\star$. Finally, we define $\tau(\alpha) = \{\alpha/_1\}$ for each $\alpha \in \Sigma$ and each $\tau \in U_1$. Now the set U_2 is obvious, while $V_2 = V_1 \cup \{A_a \mid a \in \Sigma\}$. □

Lemma 6.2. *Let K be a family of fuzzy languages that is closed under union with languages from* SYMBOL. *If $K \supseteq$* SYMBOL, *then for each fuzzy context-free K-grammar $G_1 = (V_1, \Sigma, U_1, S)$, there exists an equivalent fuzzy context-free K-grammar $G_2 = (V_2, \Sigma, U_2, S)$ such that U_2 is a singleton set.*

Proof. Let $U_1 = \{\tau_1, \ldots, \tau_m\}$ for some m ($m \geq 2$). For each k ($1 \leq k \leq m$), we define an isomorphism $i_k(\alpha) = \alpha_k$ ($\alpha \in V_1$); all α_k's are new distinct symbols such that $j \neq k$ implies that $i_j(V_1)$ and $i_k(V_1)$ are disjoint alphabets.

Define $V_2 = V_1 \cup \{i_k(\alpha) \mid \alpha \in V_1,\ 1 \leq k \leq m\}$ and $U_2 = \{\tau_0\}$ with

$\tau_0(\alpha) = \{\alpha/_1, \alpha_1/_1\}$	$\alpha \in V_1$, $\alpha_1 = i_1(\alpha)$
$\tau_0(\alpha_k) = \{\alpha_k/_1, \alpha_{k+1}/_1\} \cup \tau_k(\alpha)$	$\alpha_k \in i_k(V_1)$, $\alpha_{k+1} = i_{k+1}(\alpha)$, $1 \leq k < m$
$\tau_0(\alpha_m) = \{\alpha_m/_1\} \cup \tau_m(\alpha)$	$\alpha_m \in i_m(V_1)$.

Then $L(G_0) = L(G)$, and hence $A_{f,m}(K) \subseteq A_{f,1}(K)$ for each m ($m \geq 1$). □

The proof of Lemma 6.2 can be simplified when we put a stronger condition on the family K, e.g., the condition that K is closed under union.

Corollary 6.3. (1) *Let K be a family that is closed under union with languages from* SYMBOL. *If $K \supseteq$* SYMBOL, *then $A_{f,1}(K) = A_{f,m}(K) = A_f(K)$ for each m ($m \geq 1$).*

(2) $\mathrm{CF}_f = A_{f,1}(\mathrm{FIN}_f) = A_{f,m}(\mathrm{FIN}_f) = A_f(\mathrm{FIN}_f)$ *for each m ($m \geq 1$).*

Proof. (1) Clearly, $A_{f,1}(K) \subseteq A_{f,m}(K) \subseteq A_f(K)$ holds for each m ($m \geq 1$). From Lemma 6.2 it follows that for each m ($m \geq 1$), $A_{f,m}(K) \subseteq A_{f,1}(K)$.

(2) follows from $\mathrm{CF}_f = A_{f,1}(\mathrm{FIN}_f)$ and Corollary 6.3(1) with $K = \mathrm{FIN}_f$. □

Lemma 6.4. *Let K be a family of fuzzy languages that is closed under union with languages from* SYMBOL. *If $K \supseteq$* SYMBOL, *then $K \subseteq A_f(K)$.*

Proof. Let L_0 with $s(L_0) \subseteq \Sigma^\star$ be a fuzzy language in K. In order to show that L_0 also belongs to $A_f(K)$, we consider the fuzzy context-free K-grammar $G = (V, \Sigma, U, S)$ with $U = \{\tau\}$, $S \notin \Sigma$, $V = \Sigma \cup \{S\}$, $\tau(S) = \{S/_1\} \cup L_0$ and $\tau(\alpha) = \{\alpha/_1\}$ for all α in Σ. Then we have $L(G) = L_0$. □

7 The Main Result

This section is devoted to the principal result of this paper (Theorem 7.1) and a few of its consequences (Corollaries 7.2 and 7.3).

Theorem 7.1. *Let K be a family of fuzzy languages that is closed under union with* SYMBOL-*languages. If $K \supseteq$* SYMBOL, *then $A_f(A_f(K)) = A_f(K)$.*

Proof. First, we show that if $K \supseteq$ SYMBOL and K is closed under union with SYMBOL-languages, then (i) $A_f(K) \supseteq$ SYMBOL, (ii) $A_f(K)$ is closed under union with SYMBOL-languages, and (iii) $A_f(K)$ is closed under isomorphism. We tacitly assume that the family K is closed under isomorphism. Now (i) directly follows from Lemma 6.4.

In order to prove (ii) and (iii), let L_0 be a fuzzy language in $A_f(K)$ generated by a fuzzy context-free K-grammar $G_1 = (V_1, \Sigma_1, U_1, S_1)$, let $\{\beta\}$ be a SYMBOL-language, and let $i : \Sigma_1 \to \Sigma_3$ be an isomorphism. We will construct fuzzy context-free K-grammars G_2 and G_3 such that $L(G_2) = L(G_1) \cup \{\beta/_1\} = L_0 \cup \{\beta/_1\}$ and $L(G_3) = i(L(G_1)) = i(L_0)$, respectively. According to Lemmas 6.1 and 6.2, we assume that $U_1 = \{\tau_1\}$, and for all α in Σ_1, $\mu(\alpha; \tau_1(\alpha)) = 1$ whereas $\mu(\omega; \tau_1(\alpha)) = 0$ for all ω in $V^\star - \{\alpha\}$.

Assuming that the symbol β does not belong to N_1 ($N_1 = V_1 - \Sigma_1$), we define the grammar G_2 by $G_2 = (V_2 \cup \{S_2\}, \Sigma_1 \cup \{\beta\}, \{\tau_2\}, S_2)$ where S_2 is a new symbol (i.e., $S_2 \notin V_1 \cup \{\beta\}$), $\tau_2(S_2) = \{S_2/_1, S_1/_1, \beta/_1\}$ and $\tau_2(\alpha) = \tau_1(\alpha)$ for each $\alpha \neq S_2$. Note that $\tau_2(S_2)$ is a crisp set. To define G_3, we first extend the isomorphism $i : \Sigma_1 \to \Sigma_3$ to the isomorphism $i : V_1 \to V_3$ by $i(A) = A$ for all

A in N_1, whereas $V_3 = N_1 \cup \Sigma_3$. Then G_3 becomes $G_3 = (V_3, \Sigma_3, \{\tau_3\}, S_1)$ with $\tau_3(i(\alpha)) = i(\tau_1(\alpha))$ for each α in V_1.

The properties (i), (ii) and (iii) enable us to apply Lemmas 6.1, 6.2 and 6.4 to the family $A_f(K)$ rather than to the family K.

Now we are ready to prove the statement of Theorem 7.1. So applying Lemma 6.4 with $A_f(K)$ instead of K, yields $A_f(K) \subseteq A_f(A_f(K))$.

To establish the converse inclusion, consider an arbitrary fuzzy context-free $A_f(K)$-grammar $G = (V, \Sigma, U, S)$. By Lemma 6.2 we may assume that U consists of a single nested fuzzy $A_f(K)$-substitution τ over the alphabet V. For each α in V, let $G_\alpha = (V_\alpha, V, U_\alpha, S_\alpha)$ be a fuzzy context-free K-grammar such that $L(G_\alpha) = \tau(\alpha)$. We assume —again following Lemma 6.2— that for each α in V, the set U_α consists of a single nested fuzzy K-substitution τ_α over V_α. By Lemma 6.1, we also assume that for each τ_α ($\alpha \in V$), we have $\tau_\alpha(\sigma) = \{\sigma/_1\}$ for each σ in V. Finally, we assume without loss of generality that all nonterminal alphabets $V_\alpha - V$ of the fuzzy context-free K-grammars G_α ($\alpha \in V$) are mutually disjoint.

Thus we have to show that $L(G) \in A_f(K)$. To this end we define the fuzzy context-free K-grammar $G_0 = (V_0, \Sigma, U_0, S_0)$ as follows.

- $V_0 = \bigcup\{V_\alpha \mid \alpha \in V\}$ (So $V \subseteq V_0$, as $V \subseteq V_\alpha$ for each $\alpha \in V$.),
- $U_0 = \{\rho_\alpha \mid \alpha \in V\}$,
- $S_0 = S_S$. (Note that $S_S \in V_S$, $V_S \subseteq V_0$, and hence $S_0 \in V_0$.)

For each nested fuzzy K-substitution τ_α over V_α, we define a corresponding nested fuzzy K-substitution ρ_α in U_0 by

$$\begin{array}{ll} \rho_\alpha(\beta) = \tau_\alpha(\beta) & \beta \in V_\alpha - V \quad (\alpha \in V), \\ \rho_\alpha(\beta) = \{\beta/_1, S_\beta/_1\} & \beta \in V, \\ \rho_\alpha(\beta) = \{\beta/_1\} & \beta \in V_0 - V_\alpha \quad (\alpha \in V). \end{array}$$

Finally, it is a tedious but straightforward exercise to verify that $L(G_0) = L(G)$, and hence the fuzzy language $L(G)$ belongs to the family $A_f(K)$. □

Corollary 7.2. $A_f(\mathrm{CF}_f) = A_f(A_f(\mathrm{FIN}_f)) = A_f(\mathrm{FIN}_f) = \mathrm{CF}_f$.

Proof. Corollary 6.3(2) and Theorem 7.1 with $K = \mathrm{FIN}_f$. □

According to Corollary 7.2 we may extend the sets $\tau(\alpha)$ ($\alpha \in V$, $\tau \in U$) in a fuzzy context-free grammar $G = (V, \Sigma, U, S)$, not only with a finite number of elements, but even with a countable infinite number, as long as the resulting sets $\tau(\alpha)$ still constitute fuzzy context-free languages over V. In this sense we are able to model the case of an infinite number of grammatical errors within the framework of fuzzy context-free grammars.

Corollary 7.3. $A_f(A_f(\mathrm{ALPHA}_f)) = A_f(\mathrm{ALPHA}_f) = \mathrm{ALPHA}_f$.

Proof. The first equality follows from Theorem 7.1 with $K = \mathrm{ALPHA}_f$. The inclusion $A_f(\mathrm{ALPHA}_f) \supseteq \mathrm{ALPHA}_f$ is a consequence of Lemma 6.4. To establish the converse inclusion, consider the fuzzy context-free ALPHA_f-grammar $G = (V, \Sigma, U, S)$. As for each $\tau \in U$ and each $\alpha \in V$, we have $s(\tau(\alpha)) \subseteq V$, it follows that $s(L(G)) \subseteq V$, i.e., $L(G) \in \mathrm{ALPHA}_f$. □

8 Algebraic Closure Properties — Preliminaries

A *closure operator* Γ on a partially ordered set X is a mapping $\Gamma : X \to X$ that is *extensive*, *monotonic*, and *idempotent*, i.e., it satisfies for all x and y in X, $x \leq \Gamma(x)$, $x \leq y$ implies $\Gamma(x) \leq \Gamma(y)$, and $\Gamma(\Gamma(x)) = \Gamma(x)$, respectively.

Now Theorem 7.1 shows that A_f is idempotent on the class of all language families satisfying the conditions of Theorem 7.1. Similarly, it follows from Lemma 6.4 that on the same class A_f is extensive. Since it is straightforward to show that A_f is also monotonic on this class (i.e., $K_1 \subseteq K_2$ implies $A_f(K_1) \subseteq A_f(K_2)$ for all such families K_1 and K_2), this means that A_f is a closure operator. Consequently, if a family K of fuzzy languages meets the conditions of Theorem 7.1, then the language family $A_f(K)$ possesses interesting algebraic closure properties as we will see in Section 9. In the present section we will recall some elementary concepts, notation and basic results.

The smallest family of fuzzy languages that satisfies the conditions of Theorem 7.1 is the family ALPHA_f. But according to Corollary 7.3, we have that $A_f(\mathrm{ALPHA}_f)$ equals ALPHA_f. However, we obtain much more interesting results, as we will see in Section 9, when we turn to less trivial families of fuzzy languages, viz. to families that include FIN_f; cf. Definition 8.5.

Apart from the families in Section 5 we need the family REG_f of regular fuzzy languages, which is defined in a way similar to its crisp counterpart.

Definition 8.1. For each alphabet Σ, the *regular fuzzy languages* over Σ are defined by:

(1) The fuzzy subsets $\varnothing$, $\{\lambda/_1\}$, and $\{\sigma\}$ ($\sigma \in \Sigma$) of $\Sigma^\star$, are regular fuzzy languages over Σ.

(2) If R_1 and R_2 are regular fuzzy languages over Σ, then so are $R_1 \cup R_2$, R_1R_2, and $R_1^\star$.

(3) A fuzzy subset R of $\Sigma^\star$ is a regular fuzzy language over Σ if and only if R can be obtained from (1) by a finite number of applications of (2).

The family of regular fuzzy languages is denoted by REG_f. □

It is a routine matter to show that each regular fuzzy language is also a fuzzy context-free language; so we have $\mathrm{REG}_f \subseteq \mathrm{CF}_f$.

The family of regular fuzzy languages is closely related to an automaton model: the so-called nondeterministic fuzzy finite automaton. Similar to the crisp case we have a characterization of REG_f by fuzzy finite automata (Proposition 8.3).

Definition 8.2. 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 finite crisp 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 $\delta : Q \times (\Sigma \cup \{\lambda\}) \to \mathbb{F}(Q)$ that satisfies the following restriction: for each q in Q, $\delta(q, \lambda)$ is a crisp subset of Q. The function δ is extended to $\hat{\delta} : Q \times \Sigma^\star \to \mathbb{F}(Q)$ as follows: for all $q \in Q$, $\hat{\delta}(q, \lambda) = \delta(q, \lambda)$ and $\hat{\delta}(q, \sigma\omega) = \bigcup\{\hat{\delta}(q', \omega) \mid q' \in \delta(q, \sigma)\}$. That means, according to (2),

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

The fuzzy language $L(M)$ accepted by the NFFA M is defined by $L(M) = \{x \in \Sigma^\star \mid \hat{\delta}(q_0, x) \cap F \neq \varnothing\}$ or, equivalently, $\mu(x; L(M)) = \bigvee\{\mu(q; \hat{\delta}(q_0, x)) \mid q \in F\}$. □

Proposition 8.3. [8] *A fuzzy language L is regular if and only if L is accepted by a nondeterministic fuzzy finite automaton.* □

In Definition 5.2 we already met the notion of fuzzy substitution. In the next definition we consider two special instances.

Definition 8.4. Let $\tau : V \to K$ be a fuzzy K-substitution on the alphabet V. If 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 $\mathrm{S\hat{u}b}(K, K') = \{\tau(L) \mid L \in K;$ τ is a fuzzy K'-substitution$\}$. A family K is *closed* under fuzzy K'-substitution if $\mathrm{S\hat{u}b}(K, K') \subseteq K$, and K is *closed under fuzzy substitution*, if K is closed under fuzzy K-substitution. □

To ensure that K is less trivial than ALPHA_f, we need the notion of fuzzy prequasoid.

Definition 8.5. A *fuzzy prequasoid* K is a normalized family of fuzzy languages that is closed under fuzzy finite substitution and intersection with regular fuzzy languages. A *fuzzy quasoid* is a fuzzy prequasoid that contains a fuzzy language L such that $c(L)$ is infinite. □

It is a straightforward exercise to show that each fuzzy [pre]quasoid includes the smallest fuzzy [pre]quasoid REG_f [FIN_f, respectively], whereas FIN_f is the only fuzzy prequasoid that is not a fuzzy quasoid; cf. [8].

Let $\Pi_f(K)$ denote the smallest fuzzy prequasoid that includes the family K. Similarly, let $\Phi_f(K)$ [$\Delta_f(K)$, $\Theta_f(K)$, respectively] be the smallest family of fuzzy languages that includes K and is closed under fuzzy finite substitutions [intersection with regular fuzzy languages, fuzzy homomorphisms, respectively]. Then for each family K, we have $\Pi_f(K) = \{\Phi_f, \Delta_f, \Theta_f\}^\star(K)$ or $\Pi_f(K) = \{\Phi_f, \Delta_f\}^\star(K)$. 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; viz.

Proposition 8.6. [8] *For each family K of fuzzy languages, we have $\Pi_f(K) = \Theta_f\Delta_f\Phi_f(K) = \Phi_f\Delta_f\Phi_f(K)$.* □

When we combine the properties related to the operators A_f and Π_f we obtain an algebraic structure that is (a special case of) the fuzzy counterpart of full AFL (full Abstract Family of Languages [11]); cf. Definition 9.4.

Definition 8.7. A *full Abstract Family of Fuzzy Languages* or *full AFFL* is a normalized family of fuzzy languages closed under union, concatenation, Kleene $\star$, fuzzy homomorphism (i.e., fuzzy ONE_f-substitution), inverse fuzzy homomorphism, and intersection with regular fuzzy languages. A *full substitution-closed AFFL* is a full AFFL closed under fuzzy substitution. $\Box$

The following characterization of full AFFL is useful; its proof in [8] is a modification of a result for crisp languages, originally established in [12].

Proposition 8.8. [8] *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., $\hat{\mathrm{Sub}}(K, \mathrm{REG}_f) \subseteq K$), and under substitution in the regular fuzzy languages (i.e., $\hat{\mathrm{Sub}}(\mathrm{REG}_f, K) \subseteq K$).* $\Box$

Actually, the notion of full AFFL reflects some of the closure properties of the family REG_f of regular fuzzy languages. More formally, we have

Corollary 8.9. [8] (1) *If K is a full AFFL, then $K \supseteq \mathrm{REG}_f$.*

(2) REG_f *is the smallest full substitution-closed AFFL.* $\Box$

9 Algebraic Closure Properties — Results

In this section we first consider some simple closure properties (Lemmas 9.1 and 9.3) before we turn to more important ones (Theorem 9.6) due to our results from Section 7.

Lemma 9.1. *Let K and K' be families of fuzzy languages such that K' is closed under union with* SYMBOL*-languages and $K \supseteq K' \supseteq$ SYMBOL. Then the family of fuzzy languages $A_f(K)$ is closed under fuzzy K'-substitution.*

Proof. Let $G = (V, \Sigma, U, S)$ be a fuzzy context-free K-grammar and let $\sigma : \Sigma \to \Delta^\star$ be a fuzzy K'-substitution. Without loss of generality we assume that Σ and Δ are disjoint.

Consider the fuzzy context-free K-grammar $G_0 = (V_0, \Delta, U_0, S)$ where $V_0 = V \cup \Delta$, $U_0 = \{\tau' \mid \tau \in U\} \cup \{\sigma'\}$ with

$\sigma'(\alpha) \;=\; \textbf{if } \alpha \in \Sigma \textbf{ then } \sigma(\alpha) \cup \{\alpha/_1\} \textbf{ else } \{\alpha/_1\}$

and for each τ in U we define

$\tau'(\alpha) \;=\; \textbf{if } \alpha \in V \textbf{ then } \tau(\alpha) \textbf{ else } \{\alpha/_1\}$.

Then for each x in $\Delta^\star$, we have $\mu(x; \sigma(L(G))) = \mu(x; L(G_0))$, i.e., $L(G_0) = \sigma(L(G))$. $\Box$

Corollary 9.2. (1) *If $K \supseteq \mathrm{FIN}_f$, then $A_f(K)$ is closed under fuzzy finite substitution.*

(2) *If K is closed under union with* SYMBOL*-languages and $K \supseteq$ SYMBOL, then $A_f(K)$ is closed under fuzzy K-substitution.*

Proof. Lemma 9.1 with $K' = \text{FIN}_f$ and $K' = K$, respectively. □

Lemma 9.3. *Let K be a fuzzy prequasoid. Then the family of fuzzy languages $A_f(K)$ is closed under intersection with regular fuzzy languages.*

Proof. Let $G = (V, \Sigma, U, S)$ be a fuzzy context-free K-grammar, and let R be a regular fuzzy language accepted by a nondeterministic fuzzy finite automaton with λ-moves (NFFA) $(Q, \Sigma, \delta, q_0, F)$; cf. Proposition 8.3.

Consider the fuzzy context-free K-grammar $G_0 = (V_0, \Sigma, U_0, S_0)$ where $V_0 = \Sigma \cup \{S_0\} \cup \{[q, \alpha, q'] \mid\ q, q' \in Q,\ \alpha \in V\}$, $U_0 = \{\sigma_0, \sigma_1\} \cup \{\tau' \mid \tau \in U\}$, with

$$\begin{array}{ll} \sigma_0(S_0) = \{S_0/_1\} \cup \{[q_0, S, q]/_1 \mid q \in F\} & q \in F, \\ \sigma_0(\alpha) = \{\alpha/_1\} & \alpha \in V_0 - \{S_0\}, \\ \sigma_1(\alpha) = \{\alpha/_1\} & \alpha \in \Sigma \cup \{S_0\}, \\ \sigma_1([q, \alpha, q']) = \{[q, \alpha, q']/_1\} \cup \{\alpha \mid q' \in \delta(q, \alpha)\} & \alpha \in V, \quad q, q' \in Q. \end{array}$$

In the latter case we have, of course, $\mu(\alpha; \sigma_1([q, \alpha, q'])) = \mu(q'; \delta(q, \alpha))$.

For each τ in U, we define the fuzzy substitution τ' over V_0 by

$$\begin{aligned} \tau'([q, \alpha, q']) = \{&[q, \alpha_1, q_1][q_1, \alpha_2, q_2] \cdots [q_{n-1}, \alpha_n, q'] \mid q_1, \ldots, q_{n-1} \in Q,\ n \geq 1, \\ &\alpha_1\alpha_2 \cdots \alpha_n \in s(\tau(\alpha))\} \cup \{[q, \alpha, q']/_1\} \cup E(\tau, \alpha, q, q'), \end{aligned}$$

for all $\alpha \in V$ and all $q, q' \in Q$, where $E(\tau, \alpha, q, q')$ is the crisp set defined by

$$E(\tau, \alpha, q, q') = \textbf{if } \lambda \in s(\tau(\alpha)) \text{ and } q = q' \textbf{ then } \{\lambda/_1\} \textbf{ else } \varnothing.$$

So for the corresponding degrees of membership we have

$$\mu([q, \alpha_1, q_1] \cdots [q_{n-1}, \alpha_n, q']; \tau'([q, \alpha, q'])) = \mu(\alpha_1 \cdots \alpha_n; \tau(\alpha)), \qquad n \geq 1.$$

Since K is a fuzzy prequasoid, it easy to show that each τ' is a nested fuzzy K-substitution over V_0. The proof that $L(G_0) = L(G) \cap R$ holds is also left to the reader. □

We now turn to more complicated closure properties for fuzzy languages.

Definition 9.4. A family K of fuzzy languages is closed under *iterated nested fuzzy substitution* if for each fuzzy language L in K over some alphabet V, and each finite set U of nested fuzzy K-substitutions over V, the language $U^\star(L)$ belongs to K, where $U^\star(L)$ is defined by

$$U^\star(L) = \bigcup\{\tau_p(\cdots(\tau_1(L))\cdots) \mid\ p \geq 0;\quad \tau_i \in U,\quad 0 \leq i \leq p\}.$$

A *full super-AFFL* is a full AFFL closed under iterated nested fuzzy substitution. □

Clearly, the notion of full super-AFFL is the fuzzy counterpart of the concept of full super-AFL, introduced in [14].

We are now ready for the main results of this section (Theorems 9.5 and 9.6).

Theorem 9.5. (1) *A family K of fuzzy languages is a full super-AFFL if and only if K is a fuzzy prequasoid and $A_f(K) = K$.*

(2) *Each full super-AFFL is a full substitution-closed AFFL.*

Proof. (1) Suppose K is a full super-AFFL. By Proposition 8.8, K is a fuzzy prequasoid; so it remains to show that $A_f(K) \subseteq K$ as the converse inclusion follows from Lemma 6.4.

Let $G = (V, \Sigma, U, S)$ be an arbitrary fuzzy context-free K-grammar. Because K is a full super-AFFL, the fuzzy languages $\{S\}$, $U^\star(S)$ and $U^\star(S) \cap \Sigma^\star$ all belong to the family K. But the latter fuzzy language equals $L(G)$. Hence $L(G) \in K$ and $A_f(K) \subseteq K$.

Conversely, let K be a fuzzy prequasoid that satisfies $A_f(K) = K$. As K is a fuzzy prequasoid, we have $\mathrm{FIN}_f \subseteq K$ and thus $\mathrm{CF}_f = A_f(\mathrm{FIN}_f) \subseteq A_f(K) = K$ by Corollary 7.2. But $\mathrm{REG}_f \subseteq \mathrm{CF}_f$ and consequently we have $K \supseteq \mathrm{REG}_f$. Corollary 9.2(2) implies that K is closed under fuzzy substitution, and by Proposition 8.8 we obtain that K is a full AFFL. Now it remains to prove that K is closed under iterated nested fuzzy substitution.

Let L_0 be an arbitrary fuzzy language in K with $s(L_0) \subseteq V^\star$ for some alphabet V, and let U be a finite set of nested fuzzy K-substitutions over V. Consider the fuzzy context-free K-grammar $G = (V \cup \{S\}, V, U \cup \{\tau\}, S)$ with $S \notin V$, $\tau \notin U$, $\tau(S) = L_0 \cup \{S/_1\}$ and $\tau(\alpha) = \{\alpha/_1\}$ for each α in V. Then $L(G) = U^\star(L_0)$, $L(G) \in A_f(K) = K$, and hence $U^\star(L_0) \in K$, i.e., K is closed under iterated nested fuzzy substitution.

(2) follows from (1) together with Corollary 9.2(2). □

Theorem 9.6. (1) *If K is a fuzzy prequasoid, then $A_f(K)$ is a full super-AFFL.*

(2) *For each arbitrary family K of fuzzy languages, $A_f\Pi_f(K)$ is the smallest full super-AFFL that includes K.*

(3) *For each arbitrary family K of fuzzy languages, $A_f\Theta_f\Delta_f\Phi_f(K)$ is the smallest full super-AFFL that includes K.*

Proof. (1) By Corollary 9.2(1) and Lemma 9.3, it follows that $A_f(K)$ is a prequasoid. Now Theorem 7.1 implies that $A_f(A_f(K)) = A_f(K)$, since each prequasoid satisfies the conditions of Theorem 7.1. Consequently, $A_f(K)$ is a full super-AFFL by Theorem 9.5(1).

(2) Let $\hat{\mathcal{A}}_f(K)$ be the smallest full super-AFFL that includes K. By the inclusion $K \subseteq \hat{\mathcal{A}}_f(K)$ and the monotonicity of both Π_f and A_f, we have $A_f\Pi_f(K) \subseteq A_f\Pi_f\hat{\mathcal{A}}_f(K)$. By Theorem 9.5(1) this yields $A_f\Pi_f(K) \subseteq \hat{\mathcal{A}}_f(K)$.

But Theorem 9.6(1) and Lemma 6.4 imply that $A_f\Pi_f(K)$ is a full super-AFFL that includes K. Hence $\hat{\mathcal{A}}_f(K) = A_f\Pi_f(K)$.

(3) follows from (2) and Proposition 8.6. □

By Theorem 9.5(1) we have that K is a full super-AFFL if and only if $\Pi_f(K) = K$ and $A_f(K) = K$. So the smallest full super-AFFL $\hat{\mathcal{A}}_f(K)$, that includes K, equals $\hat{\mathcal{A}}_f(K) = \bigcup\{w(K) \mid w \in \{\Pi_f, A_f\}^\star\}$ or, written equivalently, $\hat{\mathcal{A}}_f(K) = \{\Pi_f, A_f\}^\star(K)$. According to Theorem 9.6(2) this infinite set of strings over the alphabet $\{\Pi_f, A_f\}$ can be reduced to the single string $A_f\Pi_f$. Of course, an analogous remark applies to Theorem 9.6(3).

Obviously, the following corollary is the counterpart of Corollary 8.9.

Corollary 9.7. (1) *If K is a full super-AFFL, then $K \supseteq \mathrm{CF}_f$.*

(2) *CF_f is the smallest full super-AFFL.*

Proof. (1) follows from Theorem 9.5(1), Corollary 7.2, the monotonicity of the operator A_f, and the fact that FIN_f is the smallest fuzzy prequasoid.

(2) is implied by (1) and Corollary 7.2. □

The converse of Theorem 9.5(2) does not hold: REG_f is a full substitution-closed AFFL [7], but it is properly included in CF_f. From Corollary 9.7 it follows that REG_f is not a full super-AFFL.

10 Concluding Remarks

First, we generalized fuzzy context-free grammars, as introduced in [20], to the concept of $\mathcal{L}$-fuzzy context-free grammar. Here $\mathcal{L}$ is a completely distributive complete lattice provided with an additional operation, rather than the real closed interval $[0, 1]$ as in [20]. Then we showed that using these $\mathcal{L}$-fuzzy context-free grammars we are able to model the case in which at each derivation step a choice from a finite number of possible grammatical errors is made. The generalization to a choice from an infinite number of possible grammatical errors is modeled by the concept of $\mathcal{L}$-fuzzy context-free K-grammar. However, from Theorem 7.1 and Corollary 7.2 it follows that in order to stay within the framework of fuzzy context-free languages the parameter K should satisfy: $\mathrm{FIN}_f \subseteq K \subseteq \mathrm{CF}_f$.

Our approach in describing grammatical errors has a global character: the right-hand side ω of a rule $A \to \omega$ may be replaced erroneously by a completely different string ω' with $\mu(\omega'; \tau(A)) < 1$. At first sight, allowing such a choice from an infinity of grammatical errors seems not very plausible. Indeed, to achieve an infinite choice, $\tau(A)$ must be infinite and so $\tau(A)$ must contain arbitrary long strings. Using a very long ω' rather than a short ω is "unlikely". Fortunately, this "unlikeliness" can be modeled adequately: we define μ in such a way that $\mu(\omega'; \tau(A))$ decreases as the length of ω' increases; cf. Example 5.5.

Nevertheless, the notion of $\mathcal{L}$-fuzzy context-free K-grammar turned to be a useful instrument in studying algebraic closure properties; cf. Sections 8–9.

These properties are very similar to those of ordinary, crisp context-free languages [14,2,3].

When we take $\mathcal{L}$ equal to a type-00 or to a type-10 lattice we are able to model the accumulation of grammatical errors in a satisfactory way: each additional error decreases the "quality" of the string that will be derived ultimately (Lemma 2.4). In this way a long sequence of "tiny mistakes" can result in something that resembles a "capital blunder"; see Examples 4.5 and 5.5.

In this paper we treated grammatical errors in a rather "macroscopic" fashion: instead of ω a quite different string ω' may have been used. For a more "microscopic" treatment of errors —viz. in terms of edit operations like deletions, insertions and changes of terminal symbols— in (fuzzy) context-free and context-sensitive language recognition we refer to [21] and [17]. Unfortunately, both these papers are restricted to a few concrete examples to point out the main ideas, whereas the extension to generally applicable results are left to the reader. More seriously, these papers are limited to the case in which errors only occur with respect to terminal symbols. So erroneously rewriting of nonterminal symbols —e.g., $S \Rightarrow AA$ or $S \Rightarrow BB$ instead of $S \Rightarrow AB$ according to a rule $S \rightarrow AB$ as in Example 4.4— is not dealt with at all. And the deletion of nonterminal symbols —e.g., $S \Rightarrow A$ or $S \Rightarrow B$ instead of $S \Rightarrow AB$— is not considered either in [17] or [21].

Finally, we list a few limitations of this paper briefly. Of course, there are other aspects of robustness that are not touched upon in this paper. We only mention the problems of undergeneration (Given a language L_0 and a grammar G for L_0, then G generates less than L_0.) and overgeneration (Now G generates too much: either L_0 is a proper subset of $L(G)$, or $L(G) = L_0$ but G gives rise to less desired additional ambiguities).

In this paper we only considered the problem of describing and generating grammatical errors by means of fuzzy grammars. In the companion paper [9] we will consider some recognition and parsing algorithms that are robust in the sense that they are able to deal with correct as well as erroneous inputs.

References

[1] A.V. Aho & J.D. Ullman: *The Theory of Parsing, Translation and Compiling — Volume I: Parsing* (1972), Prentice-Hall, Englewood Cliffs, NJ.

[2] P.R.J. Asveld: *Iterated Context-Independent Rewriting — An Algebraic Approach to Formal Languages* (1978), Ph.D. Thesis, Dept. of Appl. Math., Twente University of Technology, Enschede, the Netherlands.

[3] 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-Verlag, Berlin, etc.

[4] 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* (1996), World Scientific, Singapore.

[5] 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.

[6] 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.

[7] P.R.J. Asveld: The non-self-embedding property for generalized fuzzy context-free grammars, *Publ. Math. Debrecen* **54** *Suppl.* (1999) 553–573.

[8] P.R.J. Asveld: Algebraic aspects of families of fuzzy languages, *Theor. Comp. Sci.* **293** (2003) 417–445.

[9] P.R.J. Asveld: Fuzzy context-free languages — Part 2: Recognition and parsing algorithms. *Theor. Comp. Sci.* **347** (2005) 191–213.

[10] G. Gerla: Fuzzy grammars and recursively enumerable fuzzy languages, *Inform. Sci.* **60** (1992) 137–143.

[11] S. Ginsburg: *Algebraic and Automata-Theoretic Properties of Formal Languages* (1975), North-Holland, Amsterdam.

[12] S. Ginsburg & E.H. Spanier: Substitution in families of languages, *Inform. Sci.* **2** (1970) 83–110.

[13] J.A. Goguen: L-fuzzy sets, *J. Math. Analysis Appl.* **18** (1967) 145–174.

[14] S.A. Greibach: Full AFL's and nested iterated substitution, *Inform. Contr.* **16** (1970) 7–35.

[15] M.A. Harrison: *Introduction to Formal Language Theory* (1978), Addison-Wesley, Reading, Mass.

[16] J.E. Hopcroft & J.D. Ullman: *Introduction to Automata Theory, Languages, and Computation* (1979), Addison-Wesley, Reading, Mass.

[17] M. Inui, W. Shoaff, L. Fausett & M. Schneider: The recognition of imperfect strings generated by fuzzy context-sensitive grammars, *Fuzzy Sets and Systems* **62** (1994) 21–29.

[18] H.H. Kim, M. Mizumoto, J. Toyoda & K. Tanaka: L-fuzzy grammars, *Inform. Sci.* **8** (1975) 123–140.

[19] G.J. Klir & B. Yuan (eds.): *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems. Selected Papers by Lofti A. Zadeh* (1996), World Scientific, Singapore.

[20] E.T. Lee & L.A. Zadeh: Note on fuzzy languages, *Inform. Sci.* **1** (1969) 421–434.

[21] M. Schneider, H. Lim & W. Shoaff: The utilization of fuzzy sets in the recognition of imperfect strings, *Fuzzy Sets and Systems* **49** (1992) 331–337.

[22] J. van Leeuwen: A generalization of Parikh's theorem in formal language theory, pp. 17–26 in: J. Loeckx (ed.): *2nd ICALP*, Lect. Notes in Comp. Sci. **14** (1974), Springer-Verlag, Berlin, etc.

[23] W. Wechler: *The Concept of Fuzziness in Automata and Language Theory* (1978), Akademie-Verlag, Berlin.