On Grammatical Errors and Other Imperfections

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

When we extend a grammar $G = (V, \Sigma, P, S)$ with additional productions to model grammatical errors we end up with some strings over Σ that do not belong completely to either $L(G)$ or to $\Sigma^\star - L(G)$. A similar situation occurs when an automaton or a recognition/parsing algorithm A has been augmented with certain transitions in order to deal with erroneous inputs: A processes strings over Σ that neither fully are in $L(A)$ nor in $\Sigma^\star - L(A)$.

To model these phenomena adequately the notions of fuzzy language, fuzzy grammar and fuzzy automaton turn out to be appropriate. A fuzzy language L over Σ is defined by its membership function $\mu_L : \Sigma^\star \to \mathcal{L}$ where $\mathcal{L}$ is a structure that is a bit more complex then the two-element set $\{0, 1\}$ used in case of ordinary or crisp languages. A fuzzy automaton [fuzzy grammar] is a weighted automaton [weighted grammar] where the weights are taken from $\mathcal{L}$.

Originally [10], $\mathcal{L}$ was taken equal to the real closed interval $[0, 1]$, but nowadays we use a structure somewhat more complex [9, 12, 8] than the semirings usually encountered in the study of weighted automata. More precisely, $\mathcal{L}$ is a completely distributive complete lattice provided with an additional operator $\star$, i.e., $\mathcal{L} = (\mathcal{L}, \wedge, \vee, 0, 1, \star)$ where $(\mathcal{L}, \wedge, \vee, 0, 1)$ is a completely distributive complete lattice and $(\mathcal{L}, \star, 1)$ is a commutative monoid; these two substructures are coupled by distributivity laws, and $0 \star a = 0$ holds for all a in $\mathcal{L}$; cf. [1, 2].

Imperfections can be small ("tiny mistakes": weights are close to, but unequal to 1) or big ("capital blunders": weights are close to, but unequal to 0). Multiple errors in a derivation or computation should be accumulated, the result of which determines the quality (the degree of perfection) of the string to be generated or accepted ultimately.

The three operations of $\mathcal{L}$ reflect several aspects of the generation or computation process:

- The $\vee$-operation plays an important rôle in the presence of nondeterminism or ambiguity.
- We need the $\wedge$-operation for imposing regular restrictions, in particular for "intersection with the set of words over a terminal alphabet".
- The $\star$-operation is essential in describing the accumulation of errors properly ("Making an error twice is worse than making it once." "A long sequence of tiny mistakes may result in something that looks like a capital blunder.")

The $\vee$-operation is needed for another reason; viz. the occurrence of an error may be compensated by another error to obtain a perfectly correct string w in the language $L(G)$ or $L(A)$. In order to yield $\mu_{L(G)}(w) = 1$ or $\mu_{L(A)}(w) = 1$, we need to take the supremum over all possible derivations or computations, respectively, including the completely correct ones.

In this framework some properties of context-free [3], regular [2, 6] and Lindenmayer [1, 6] languages have been generalized to their "fuzzified" counterparts. For instance, context-free grammars augmented with rules that model grammatical errors give rise to modifications of well-known parsing algorithms [4] which, apart from their usual job, are able to deal with erroneous inputs as well.

Although many of these generalizations hold for arbitrary $\mathcal{L}$, some results have been only established for the case in which $\mathcal{L}$ is linearly ordered (of which the real closed interval $[0, 1]$

with operations min, max and multiplication is the most frequently used, practical example).

As an illustration we consider the problem of (in)dependency of operations on fuzzy languages. An operation o from a class C of operations on fuzzy languages is called *independent* if o cannot be expressed as a polynomial over $C - \{o\}$. A major tool in studying independency is the set of algebraic closure operators that correspond to the subsets of C. This set turns out to be a bounded partially ordered monoid, the structure of which (i.e., its Cayley table and its partial order) answers many (in)dependency questions. In determining the structure of this monoid, it is crucial to establish (i) defining relations, and (ii) inequalities. Now (i) is usually possible for arbitrary $\mathcal{L}$, whereas (ii) relies on translating a proper inclusion between families of crisp languages to a proper inclusion between the corresponding families of fuzzy languages. In this translation process the linear order of $\mathcal{L}$ is probably indispensable [5, 7].

On the other hand, when $\mathcal{L}$ is not linearly ordered —even in the simplest case (i.e., $\mathcal{L}$ is the four-element distributive lattice that is not a chain, while $\star$ coincides with $\wedge$)— unexpected matters will occur: there exists two fuzzy context-free languages L_1 and L_2 over Σ with their crisp parts, defined by $c(L_i) = \{w \in \Sigma^\star \mid \mu_{L_i}(w) = 1\}$, being empty (i.e., $c(L_i) = \varnothing$ for $i = 1, 2$), whereas the crisp part of their union $c(L_1 \cup L_2)$ is not context-free; cf. [11, 4].

1. 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.
2. P.R.J. Asveld, The non-self-embedding property for generalized fuzzy context-free grammars, *Publ. Math. Debrecen* **54** *Suppl.* (1999) 553–573.
3. 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.
4. P.R.J. Asveld, Fuzzy context-free languages — Part II: Recognition and parsing algorithms (2000), CTIT Techn. Rep. No. 00-25, Enschede, the Netherlands.
5. P.R.J. Asveld, Some operators on families of fuzzy languages and their monoids, pp. 1–14 in: M. Ito, Gh. Păun & Sh. Yu (eds.), *Words, Semigroups, and Translations* (2001), World Scientific, Singapore.
6. P.R.J. Asveld, Algebraic aspects of families of fuzzy languages, *Theor. Comp. Sci.* (to appear).
7. P.R.J. Asveld, Independency of operations on fuzzy languages, [in preparation].
8. G. Gerla, Fuzzy grammars and recursively enumerable fuzzy languages, *Inform. Sci.* **60** (1992) 137–143.
9. H.H. Kim, M. Mizumoto, J. Toyoda & K. Tanaka, *L*-fuzzy grammars, *Inform. Sci.* **8** (1975) 123–140.
10. E.T. Lee & L.A. Zadeh, Note on fuzzy languages, *Inform. Sci.* **1** (1969) 421–434.
11. M. Mizumoto, J. Toyoda & K. Tanaka, Examples of formal grammars with weights, *Inform. Process. Lett.* **2** (1973) 74–78.
12. W. Wechler: *The Concept of Fuzziness in Automata and Language Theory* (1978), Akademie-Verlag, Berlin.