Relja Vulanović

On Measuring Language Complexity as Relative to the Conveyed Linguistic Information

Abstract

In this mathematical approach to language complexity, a previously proposed formula for measuring grammar complexity is derived in a different way and somewhat modified. The formula measures relative language complexity, "relative" because the conveyed linguistic information is taken into account. Many examples, either abstract or representing structures of natural languages, are used in the derivation and to illustrate the method.[1]

1. Introduction

Language complexity has recently become very interesting to researchers, as witnessed by an increasing number of conferences and publications on the topic. In this paper, I present a newly modified formula for measuring grammar complexity. The formula is based on a formal grammatical model and language complexity is identified with the complexity of the formal grammar. Only simple sentences are modeled, but with various grammatical structures. The formula enables a comparison of the complexity of these structures relative to the linguistic information they convey. Simply put, a grammar is less complex, and at the same time more efficient, if it conveys more information (more meaning) with fewer forms and rules. As opposed to this kind of information-relative complexity, absolute complexity only takes forms and rules into account. This can be illustrated by McWhorter's (2001: 127) discussion of Kikongo and Japanese. In Kikongo, there are

[1] As a mathematician, I enjoy and appreciate any opportunity to share my work with linguistic community. I presented an earlier version of this paper at the conference on *Approaches to Complexity in Language*, Helsinki, August 24–24, 2005. That this is now a much improved version, I am grateful to two anonymous SKY referees. My thanks are also due to Leena Kolehmainen, without whose encouragement this paper would not be in its present form.

four kinds of past tense (including completive), while Japanese has only one past tense and no grammaticalized indicator of completiveness exclusively. This is why McWhorter considers this part of the Kikongo grammar more complex than the corresponding part of Japanese. My formula for absolute grammar complexity supports this. However, speaking relatively, Kikongo cannot be classified as more complex just because it has more tenses, since each tense means a different kind of linguistic information. This information is not important in Japanese, but this does not imply that it is not important at all. For whatever reason, it is important to the speakers of Kikongo. The need for conveying more information justifies the use of more tenses in Kikongo, thus information-relative complexity of Kikongo is not necessarily greater than that of Japanese. It is one of the goals of the present paper to point out that the conveyed linguistic information should be taken into account when measuring language complexity. The purpose of the last example in the paper is to illustrate this in particular.

The simpler the grammar, the greater its efficiency. This is why I define the measures of grammar complexity and efficiency as reciprocal to each other. My concept of grammar efficiency, which goes back to Vulanović (1991), is inspired by machine efficiency. Machine efficiency can be defined as the measure of the useful output divided by the measure of the input. It is energy or work that is measured in physics and engineering, but if grammars are also considered "machines," then linguistic input and output have to be measured. The information conveyed by the grammar is viewed as machine output and the forms and rules as machine input. This is the main idea behind the way the complexity formula is constructed. Since my approach is theoretical and model-based, once the formal grammatical model is established as a framework, I do not consider anything outside the model. Therefore, the question is how to define and measure grammatical complexity within the model. Moreover, the derived formula becomes the definition of complexity in the model. This perhaps is a luxury that only mathematicians enjoy, but working with a model enables utilization of mathematical formality and precision.

The first grammar efficiency formula (Vulanović 1991) is very elementary and it is developed further in Vulanović (1993, 2003). A somewhat modified formula is derived in a different way in the present paper. The derivation and the whole presentation is simpler and less mathematical than in Vulanović (2003), making the results more easily understandable to linguists. This is another goal of the present paper. Unlike in Vulanović (2003), the formula for measuring information-relative complexity is intro-

duced here step by step. I start with a very simple first version of absolute complexity and then use a sequence of examples that motivate more sophisticated versions until the final one is reached. The final formula takes processing difficulty and ambiguity into account, as well as the amount of the information that the modeled grammar conveys. During the derivation of the formula, I compare it to the complexity criteria by other authors. My intention is not to model everything these criteria state, but my formula agrees with the criteria on many points. One of the most significant differences is in the treatment of syntactic, semantic, and pragmatic functions (SSPFs). Several authors consider them complexifying factors, but in my model, only forms and rules are complexifying factors and SSPFs constitute the grammar output.

The rest of the paper is organized as follows. In section 2, I survey and comment on some existing results on language complexity. I introduce in section 3 the notation and a description of the formal grammatical model which serves as the basis for the formulas. The derivation of the information-relative complexity formula is presented in section 4. Section 5 contains further examples. Finally, section 6 offers some concluding remarks.

2. Language complexity according to other authors

In his book, Dahl (2004) discusses many factors which are important for linguistic complexity. To him (Dahl 2004: 25), relative complexity means the length of the *additional* description necessary to characterize some entity within a given theory. The theory already provides some information about the entity and its description does not have to contain this information. If no background information is assumed, the length of the *full* description of the entity would be its absolute complexity. Following this, Miestamo (2006a, 2006b, to appear) speaks of absolute complexity as theory-oriented, as "the number of parts in a system" or "the length of the description of a phenomenon" (Miestamo 2006a). However, he considers relative complexity the kind of language complexity discussed in Kusters (2003), where language processing/acquisition/learning difficulty is taken into account. This, therefore, means complexity relative to language users. My concept of relative complexity is something different—complexity relative to the conveyed information. In order to avoid any possible confusion, I refer to it as the *information-relative complexity*.

Hawkins (1994, 2004) considers language complexity without differentiating between absolute and relative complexity. His view of complexi-

ty is based on the ideas in Miller and Chomsky (1963) and Frazier (1985). Miller and Chomsky define syntactic complexity as the ratio of the number of non-terminal nodes to the number of terminal nodes in the phrase-structure tree. Frazier modifies this metric by making the node count local, i.e. by considering not the whole sentence but groups of terminal nodes and the nodes dominating them. Hawkins' (1994) theory of Early Immediate Constituents uses a refined version of this local metric. In Hawkins (2004), he extends his work beyond phrase-structure nodes to include morphology, morphosyntax, and semantics. He states that

(1) *Complexity increases with the number of linguistic forms and the number of conventionally associated (syntactic and semantic) properties that are assigned to them when constructing syntactic and semantic representations for sentences.* (Hawkins 2004: 9)

However, in his 2004 book, Hawkins is concerned more with grammatical efficiency than complexity, and to him, the measures of the two are not reciprocals of each other:

(2) *Efficiency* (…) *may involve more or less complexity, depending on the syntactic and semantic representations to be assigned to a given sentence and on their required minimum of complexity* (…) *some structures can be more efficient than others relative to this minimum.* (ibid.)

Hawkins proposes three general principles of efficiency, which are described as preferences of the human processor. The principles are given in (3).

(3) Hawkins' (2004) efficiency principles.
   i. Minimize Domains: *The human processor prefers to minimize the connected sequences of linguistic forms and their conventionally associated syntactic and semantic properties in which relations of combination and/or dependency are processed.* (Hawkins 2004: 31)
   ii. Minimize Forms: *The human processor prefers to minimize the formal complexity of each linguistic form F (its phoneme, morpheme, word, or phrasal units) and the number of forms with unique conventionalized property assignments, thereby assigning more properties to fewer forms.* (Hawkins 2004: 38)
   iii. Maximize On-line Processing*: The human processor prefers to maximize the set of properties that are assignable to each item X as X is processed, thereby increasing On-line Property to Ultimate Property ratios.* (Hawkins 2004: 51)

Obviously, the kind of complexity Frazier and Hawkins consider is processing complexity, which some authors, like McWhorter (2001: 134), Dahl (2004: 39), and Miestamo (to appear: 8) feel should not be part of complexity metric. They view processing complexity as user-related and unsuitable to be part of an objective, information-theoretic concept of complexity. My opinion is that processing can be defined in an objective, theoretical way and can be analyzed as such. The metric I present here is theoretical and it has a component which depends on processing (although, contrary to Hawkins, I am not interested in the human processor). The number of parts in a system does not tell us what these parts are and how they function in the system. Similarly, judging the complexity of different phenomena based on the length of their descriptions seems too simplistic to me. We do not know what these descriptions contain nor do we know the relationship between the components that the phenomenon consists of. Juola's (1998) computerized data-compression approach may represent the characteristics of the system better than the number of system parts, although those characteristics are only implicitly included in the measurements.

It seems inevitable to have to restrict studies of complexity from *global complexity* to *local complexity* (terms used by Miestamo (to appear)). Global complexity is the overall language complexity which requires a complete and detailed grammar. As Miestamo (2006a) points out, this is a formidable task for which we do not have adequate linguistic tools. He refers to this problem as the problem of representativity. What we can accomplish is studies of complexity of separate aspects (local areas) of grammar across languages. For instance, Miestamo (2006b) analyzes the complexity of standard negation while Nichols (1992) and Juola (1998) discuss morphological complexity. It, therefore, should not be surprising that my model only represents simple structures, although the same approach can be applied in principle to more complicated constructions. Describing the local grammatical area considered in this paper, I can say that it deals mainly with the complexity of word order and cases in simple sentences. Many other language features (like agreement and cross-reference, discontinuous constituents, pro-drop constructions, stylistic nuances, etc.) are not intended to be represented in the model. Nevertheless, the model is still capable of describing various grammatical structures, as many examples in the paper show. And, if we do not know how to measure complexity of these simple structures, how can we hope to accomplish this with the more complicated ones?

I will compare below my metric to the works mentioned above and particularly to McWhorter (2001, to appear b). McWhorter (2001) is the leading paper in a discussion of language complexity, to which the whole double issue of *Linguistic Typology*, 5:2/3 (2001), is devoted. In this paper, McWhorter proposes the following metric (McWhorter 2001: 135–137):

(4) A grammar is more complex than another to the extent of
   i. marked members of its phonemic inventory,
   ii. rules its syntax has to process,
   iii. fine-grained semantic and/or pragmatic distinctions it gives overt and grammaticalized expression to,
   iv. its use of inflectional morphology.

He modifies the above criteria in McWhorter (to appear b). This is how they are described in McWhorter (to appear a: 2):

(5) A grammar is more complex than another to the extent of its
   i. overspecification ("marking of semantic categories left to context in many or most languages, such as evidential marking"),
   ii. structural elaboration ("number of rules mediating underlying forms and surface forms, such as morphophonemics"),
   iii. irregularity.

3. The formal grammatical model

An elementary formal grammar is used to model simple sentences. It is a version of Dik's (1997) Functional Grammar (FG), that I have used in Vulanović (2005), but I will present it this time in analytic form. I find FG convenient for the purpose of this research because it does not involve phrase-structure trees and the underlying structures that should be ordered (linearized) are directly accessible. Languages use word order to avoid ambiguity and word order is related to the parts-of-speech system (Hengeveld et al. 2004). This is included in the present formal grammar.

Only core predications of FG are modeled, with terms and satellites treated equally. The lexicon, phonology, term formation, and the FG states of affair are not represented. Only verbal predicates with fully expressed nominals are considered. This is illustrated below by the English sentence

(6) Mary was bought a book by John.

The structure of this sentence can be represented as

(7) N Pas N [by N],

where N stands for nouns (or even some noun phrases since 'a book' is represented simply by an N), Pas for the used passive form of the verb 'buy' (without explicit indication of the person, number, or tense), and where the square brackets mean that the phrase 'by N' is a single unit in the string of symbols in (7). When (7), which is considered a sentence, is parsed, strings of the FG semantic, syntactic, and pragmatic functions (abbreviated together as SSPFs) are assigned to each component of (7) and the result is the analysis

(8) N Pas N [by N] → BenSubj P Go Ag.

In (8), Ben (beneficiary), Go (goal), and Ag (agent) are semantic functions and Subj (subject) is a syntactic function. Obj (object) is the only other syntactic function in FG (it does not have to be assigned in this example), but I find it convenient here to consider P (predicate) a syntactic function too. Ben and Subj are joined together to form a string of SSPFs which is assigned to the first N in (7), i.e. to 'Mary' in (6). Pragmatic functions are not assigned in (8), nor will they be needed in any example of this paper.

The analysis in (8) can be viewed as the result of two components of the grammar. The first one is a mapping describing individual assignments of the SSPFs, which in this example looks like

(9) N → BenSubj, Go, [by N] → Ag, Pas → P.

Based solely on the mapping in (9), sentence (7) would be ambiguous since it would have two analyses: BenSubj P Go Ag and Go P BenSubj Ag. The second component of the grammar is needed to provide the permissible orders of the SSPFs. If the order BenSubj P Go Ag is the only one permitted, (7) has to be analyzed like in (8). Note, however, that many other permutations of (7) can be analyzed unambiguously as long as the relative order of BenSubj and Go is fixed. If, like in (8), BenSubj precedes Go, there are 4!/2 = 24/2 = 12 other unambiguous sentences, like Pas[by N]NN for instance.

Generalizing and formalizing the above example, we can describe a grammar as a mapping Φ of type (9) and a set R of permissible orders of

SSPFs. On the left side of each arrow in Φ, there is a single element of a set *C* symbolizing word classes, case forms, verbs and their forms, i.e. any grammatical category that is used to convey the strings of SSPFs occurring on the right side of each arrow. Those strings are elements of another set, *F*. Note that each element of *C* occurs exactly once in Φ, whereas on the right side of each arrow there may be more than one element of *F*. This models possible violations of the One-Meaning–One-Form principle, which is important for the discussion of grammatical complexity (Miestamo, to appear).

Let *C*, *F*, and *R* have *k*, *n*, and ρ elements respectively.

From this point on, the notation for SSPFs is simplified. S and O are used instead of XSubj and XObj respectively, where X is any semantic function. O also replaces SSPF strings starting in Go, except for GoSubj (which is rendered as S). This brings the notation closer to works on word order ty-pology, which usually refer to S, O, and V. P is, nevertheless, preserved in-stead of V, since V, unlike S and O, is not an SSPF.

The following four examples illustrate the above further. In all of them, $F = \{S, O, P\}$ (thus $n = 3$) and verbs (V) are used as a grammatical category conveying P.

Example 1. Let $C = \{N, V\}$, so that $k = 2$. N is interpreted as either S or O:

(10) Φ: N → S, O, V → P.

Furthermore, let

$R = \{SOP, SPO, PSO\}$.

This grammar, denoted by G_1, admits three sentences (NNV, NVN, and VNN), which are all unambiguous. If another string is added to *R*, some sentences become ambiguous, e.g. if OSP is an additional string in *R*, NNV is an ambiguous sentence since it can be interpreted as both SOP and OSP. Therefore, if ρ^* denotes the greatest possible number of orders in *R*, so that no sentence is ambiguous, then in this example, $\rho = \rho^* = 3$.

In the case of standard transitive English sentences, $\rho = 1 < \rho^*$, the only string in *R* being SPO. Let G_E denote this grammatical structure.

Let us also calculate the quantity ρ' which will be needed in section 4. ρ' represents the total number of all parses attempted when each permutation of each possible sentence is parsed. It is assumed here that the parsing

process is only based on the information obtained from mapping Φ and set *F*, and not from set *R*. The reason for this will be explained later in example 6. Because of this assumption and because all permutations are considered, more orders may have to be analyzed than what is contained in set *R*. It is also assumed that parsing proceeds from left to right, one element of set *C* at a time. For instance, using the mapping in (10), sentence NNV can be parsed in four ways: SOP, OSP, SSP, OOP. However, the parser has the information from set *F* that each sentence has to convey S, O, and P, and this is why the last two parses are unacceptable. In other words, if, say, S is assigned to the first N in the sentence, then O has to be assigned to the second N. There will be no attempt to parse NN as SS or OO at all. In this case, there are two parsing attempts, starting in SO and OS. Both can be completed successfully, but, in general, it is the count of all attempts, successful or not, that is used to form ρ'. Since the other two permutations, NVN and VNN, are parsed analogously, there are six attempted parses in all and the value of ρ' is set to equal 6.

The idea behind ρ' is to measure how much the One-Meaning–One-Form principle is violated. The greater the extent of the violation, the greater processing difficulty. Thus, ρ' represents here processing difficulty, which is understood very formally. I have no intention of connecting this in any way with how the human parser operates. In this example, ρ' happens to be equal to $n! = 3! = 6$, but in general, ρ' is not the same quantity as $n!$. Some other examples will show this. The present definition of ρ' is new. The quantity introduced in Vulanović (2003) can be used as well and it is easier to calculate. However, the new ρ' is connected better to the parsing process.

Example 2. This example models simple transitive sentences in languages with object marking. Set *C* has 3 elements ($k = 3$): the nominative case (Nom) is used to convey S, the accusative case (Acc) to convey O, and V conveys P. Thus,

(11) Φ: Nom → S, Acc → O, V → P.

This time, ρ^* is 6 since all six permutations of S, O, and P can be included in *R* without creating ambiguity. The grammar with the mapping in (11) and $\rho = \rho^* = 6$ is denoted by G_2. It is easy to see that in this grammar the value of ρ' is also 6.

Example 3. Consider a grammar in which S and O are coded on the verb. There is only one nominal form, N, but there are two verbal forms, V^+ and V^-:

(12) Φ: $N \rightarrow S, O, \quad V^+ \rightarrow P, \quad V^- \rightarrow P.$

V^+ indicates that S precedes O and V^- means the opposite direction. This is the first example in which the same SSPF is assigned to different elements of set *C*. Whenever this happens, the whole corresponding pair of elements from *C* and *F* is used to represent the SSPF when the orders in set *R* are formed. Thus, in this example the pairs are (V^+, P) and (V^-, P). These ordered pairs are necessary since by referring to P alone it would be impossible to describe the orders in set *R*. *R* contains the following 6 orders:

$$R = \{SO(V^+, P), S(V^+, P)O, (V^+, P)SO, OS(V^-, P), O(V^-, P)S, (V^-, P)OS\}.$$

This is the maximum possible number of orders in *R* without permitting ambiguous sentences, thus $\rho = \rho^* = 6$. Let G_3 denote this grammar.

$\rho' = 12$ for the following reason. There are three permutations of NNV^+ and three permutations of NNV^-. Each permutation requires two parsing attempts, like in example 1. Note that although V^+ means that S precedes O, this cannot be concluded from mapping (12) or set *F*. This information is available in set *R* which is not used in parsing. Ultimately, six parsing attempts are unsuccessful since the resulting orders are not in *R*.

This is an abstract example, but it will be used to motivate one step in the derivation of the efficiency formula. Moreover, similar constructions can be found in Algonquian languages. In discourse segments of medium size, Cree (Wolfart and Carroll 1981) uses the unmarked proximate form for the more central third person, whereas all other third persons are in the marked obviative form. The agent in a sentence may be in either proximate or obviative form without having any additional marking. Word order is not used either to specify the agent, rather, it is the verbal category of direction that carries this information. One verbal form indicates that the action is from proximate to obviative and another signifies the opposite direction.

Example 4. Returning to a structure similar to G_2, let us suppose that there are two declensional noun classes with different nominative and accusative forms. This is represented in mapping Φ below:

$$\Phi: \quad Nom_1 \rightarrow S, \quad Nom_2 \rightarrow S, \quad Acc_1 \rightarrow O, \quad Acc_2 \rightarrow O, \quad V \rightarrow P.$$

In this case, $k = 5$ and the orders in R can be described simply by referring to the SSPFs, like in example 2. However, because of the convention introduced in example 3, instead of S and O, the corresponding *C*-*F* pairs are used. Let

$$R = Per\{(Nom_1,S), (Acc_1,O), P\} \cup Per\{(Nom_1,S), (Acc_2,O), P\} \cup Per\{(Nom_2,S), (Acc_1,O), P\} \cup Per\{(Nom_2,S), (Acc_2,O), P\},$$

where *PerA* denotes the set of all permutations of all elements in a set *A*. In this grammar, denoted by G_6, $\rho = \rho^* = 24$. The 24 orders correspond to the 24 unambiguous sentences, obtained by six permutations of each Nom_iAcc_jV for $i = 1, 2$ and $j = 1, 2$. The value of ρ' is also equal to 24. Like in example 2, so here, $\rho' = \rho^*$. This is always the case when each element of set *C* conveys exactly one SSPF.

In the grammatical model described above, SSPFs are conveyed by the elements of set *C* and by word order represented in set *R*. This is why throughout my work I refer to the elements of *C* and word order as *(grammatical) conveyors*. What they convey I call *linguistically relevant information*, or simply *linguistic information*. This is not necessarily just the SSPFs, like in the previous examples, but any other information that sentences of a language have to convey. Tense, for instance, is considered in example 13 as this kind of information. There is a great variation across languages in terms of what linguistic information they consider necessary to convey. From the point of view of mathematical formalism, it suffices to say that anything that is placed in set *C* is a "grammatical conveyor" and anything in set *F* is "linguistic information." We may think of set *C* as of a set of linguistic categories or forms and of set *F* as of their meanings. Syntactic rules are contained in set *R*.

4. From absolute to information-relative language complexity

A grammar complexity formula, based on the above grammatical model, is derived in this section. It seems reasonable to start the construction of the

formula from grammatical conveyors. As for the conveyors in set *C*, their greater number implies greater complexity. This is in agreement with (1), which states that complexity increases with the number of linguistic forms. Criteria (5i) and (5iii) imply the same (among many other things). The rationale behind criterion (4iv) is that inflection is usually a complexifying factor because of its effects upon a grammar over time and "the fact that some inflection, such as gender marking and declensional noun classes, does not correspond to concepts expressed by all grammars, but is instead purely supplementary to a grammar's machinery" (McWhorter 2001: 138). As example 4 shows, declensional noun classes increase the number of elements in *C*.

Word order, as a conveyor, involves more rules if it is less free, i.e. if there are fewer elements in *R*. Gil (2001: 344) also treats the free word order of Riau Indonesian as a feature indicating a less complex grammar. McWhorter (2001) does not dwell on word order too much, except when talking about word order in questions in Tsez and Saramaccan. However, his criteria (4ii) and (5ii) indicate that complexity increases with the number of rules, which certainly include word order rules as well.

Therefore, complexity (here, I start identifying complexity with the formula under development) is directly proportional to k and indirectly proportional to ρ. Probably the simplest formula of this kind is

$$\text{(13)}\quad \text{AC}'' = \frac{k}{\rho},$$

where AC stand for *absolute complexity*, while " indicates that this definition is two steps away from the final formula for AC.

Let AC''_i, $i = 1, 2, 3, 4$, denote the values of AC" for the corresponding grammars G_i of the previous section. It is easy to see that

$$\text{AC}''_1 = \frac{2}{3}, \quad \text{AC}''_2 = \text{AC}''_3 = \frac{3}{6} = \frac{1}{2}, \quad \text{and} \quad \text{AC}''_4 = \frac{5}{24}.$$

Simple transitive English sentences are even more complex since for this structure, $\text{AC}''_\text{E} = 2/1 = 2$.

It is immediately clear that the above results are unacceptable. G_4 is obviously more complex than G_2, but $\text{AC}''_4 < \text{AC}''_2$. Moreover, the values of AC''_2 and AC''_3 are equal, even though the G_2 mapping (11) looks

simpler than (12) in G_3. Intuitively, it even seems that G_1 is simpler than G_3, but $AC''_1 > AC''_3$. Therefore, the formula for AC should be modified so that these inconsistencies can be resolved.

One of the problems with AC'' is that greater values of k often imply greater values of ρ, like in example 4. This should be compensated for by a new factor inserted on the right-hand side of (13). This new factor should also depend on mapping Φ, which reveals how the conveyors in set C are used and whether the One-Meaning–One-Form principle is preserved or not. In case of the latter, like in (10) and (12), sentence processing is more difficult and greater complexity should be assigned to such structures. All of the above is covered by the number of attempted parses ρ', the quantity already evaluated for the grammars in examples 1–4. Therefore, the next modification of the measure of AC is

$$(14)\quad AC' = \frac{\rho'}{\rho} \cdot k\,.$$

Factor ρ' makes an important difference between the four grammars of the previous section because

$$(15)\quad AC'_1 = \frac{6}{3} \cdot 2 = 4\,,\quad AC'_2 = \frac{6}{6} \cdot 3 = 3\,,\quad AC'_3 = \frac{12}{6} \cdot 3 = 6\,,\text{ and }\quad AC'_4 = \frac{24}{24} \cdot 5 = 5\,.$$

This corrects the previous problem since $AC'_1 < AC'_3$, $AC'_2 < AC'_3$, and $AC'_2 < AC'_4$. As for G_E, its complexity is now estimated by $AC'_E = (6/1)2 = 12$. In G_4, the ρ' factor simply cancels out the seemingly artificial increase of $\rho = \rho^*$, whereas in G_1 and G_3, it modifies the complexity measure by taking violations of the One-Meaning–One-Form principle into account.

In order to continue with the derivation of the complexity formula, we have to introduce another example.

Example 5. Consider G_1 again but assume now that all six permutations of S, O, and P are in R. In this grammar, denoted by G_5, ρ is increased to $\rho = 6$ and $AC'_5 = (6/6)2 = 2$, which means that, according to (14), G_5 is less complex than G_1 although every sentence in G_5 is ambiguous.

The above example shows that AC' is still not an adequate measure of grammar complexity. It should not be possible to decrease grammar com-

plexity by permitting more ambiguous sentences. This is not to say that *every* grammar without ambiguity is less complex than an ambiguous one, since the metric for grammar complexity also involves other factors. However, if the only difference between two grammars is the amount of ambiguity, then it is reasonable to declare the more ambiguous grammar more complex. This is not so in example 5 and this is why (14) should be modified further. The formula below takes care of the problem illustrated by example 5. It is the last iteration in the process of deriving a reliable metric for absolute grammar complexity:

$$(16)\quad \mathrm{AC} = \gamma\, k \quad \text{with} \quad \gamma = \frac{\rho'}{\rho - \rho_0}.$$

Here, ρ_0 measures ambiguity and if it is greater, the measure of complexity is greater. If there is no ambiguity, $\rho_0 = 0$ and then AC = AC'. The quantity ρ_0 is defined as

$$\rho_0 = \sum_{i=1}^{\rho} \frac{a_i}{s_i},$$

where s_i is the length of the ith string (order) in R and a_i indicates how many components of that string are analyzed ambiguously. In G_5, each of the six strings in R has three components (i.e. $s_i = 3$, $i = 1, 2,..., 6$), two of which give rise to ambiguity. Therefore, $\rho_0 = 6(2/3) = 4$, which implies that $AC_5 = [6/(6 - 4)]2 = 6$. This puts complexity measures of G_5 and G_1 in the right relation, $AC_5 = 6 > AC_1 = 4$.

The present formula for ρ_0 is a slight refinement of the previous one in Vulanović (2003), which simply counts all ambiguous sentences ignoring how many words can still be analyzed unambiguously. If all components of all strings in R have ambiguous interpretation, then $\rho_0 = \rho$ and AC in (16) is understood as infinite complexity. This is the main reason for the way ρ_0 is included in formula (16). $\rho'\rho_0/\rho$, for instance, has the same effect as the adopted $\rho'/(\rho - \rho_0)$, in that complexity increases together with ambiguity, but $\rho'\rho_0/\rho$ does not become infinite when $\rho = \rho_0$. If a language, as described by the present model, has infinite complexity, this basically indicates that it is useless—no information can be deduced from its sentences.

Frazier (1985: 135) recognizes ambiguity as a source of processing complexity. Hawkins (2004) discusses ambiguity issues related to his efficiency principles (3), particularly to (3ii) Minimize Forms. As the number of forms is reduced, "[c]hoices have to be made over which properties get priority for unique assignment to forms, and the remaining properties are then assigned to more general forms that are ambiguous, vague, or zero-specified with respect to the property in question" (p. 38). Other afore-mentioned references on language complexity do not deal with ambiguity that much. This is not surprising since there are many other factors, beyond syntax and morphology, that language can use to resolve ambiguity—Hengeveld et al. (2004) mention prosodic, semantic, pragmatic, and visual factors. However, in a simple theoretical model like the present one, ambiguity plays a significant role, as illustrated by example 5. Note also that the model does not attempt at representing all possible types of ambiguities, but only the structural ones that can typically be resolved by restricting word order. Another thing to be noted is that ambiguity, as well as the whole complexity measure, are evaluated *within* the grammar. Native speakers of Kikongo, to use the example from the introduction, would probably find Japanese past tense ambiguous, but ambiguity of Japanese past tense is measured based on the requirements of Japanese grammar, not that of Kikongo.

Since there is no ambiguity in grammars G_1, G_2, G_3, and G_4, their AC values remain the same as the AC' values in (15). According to this, G_2 is the least complex grammar of the three. However, why should G_1 be more complex than G_2? It uses one less conveyor and therefore cannot permit completely free word order without creating ambiguity. It is impossible to achieve a smaller value of AC with only two conveyors. In this sense, G_1 is an optimal grammar when $n = 3$ and $k = 2$, and so is G_2 when $n = 3$ and $k = 3$. Therefore, the two grammars should be equally complex. This already means information-relative grammar complexity since it is analyzed here how complex the grammatical structure is in comparison to an optimal structure. The optimal grammatical structure uses the same number of conveyors and conveys the same information, but has the smallest possible value of AC. The formula for information-relative grammar complexity, IRC, can be derived by scaling AC,

$$\mathrm{IRC} = w' \gamma k,$$

where w' is a weight determined so that IRC = 1 when the grammatical structure is optimal. It is convenient to write w' as $w' = w/n$, which gives the final formula for measuring grammar complexity,

$$\text{(17)}\quad \text{IRC} = w\gamma \cdot \frac{k}{n} = w \cdot \frac{\rho'}{\rho - \rho_0} \cdot \frac{k}{n}.$$

This formula is essentially the same as the one in (Vulanović 2003).

Let $\Gamma_{k,n}$ denote the class of grammars that all have k conveyors in set C and n SSPFs in set F. Then the formal definition of an optimal grammar is related to the following problem:

> Maximize AC. Within the class $\Gamma_{k,n}$ of grammars, find a mapping Φ and set R so that each element of F appears exactly once in Φ, no sentence is ambiguous ($\rho_0 = 0$), and γ has the greatest possible value.

This problem is not always solvable. For instance, if $k > n$, Φ cannot be constructed as required. But, if there is a solution (which does not have to be unique), then this solution is an optimal grammar which is considered to have the least possible amount of complexity in $\Gamma_{k,n}$. The IRC measure of complexity of this grammar is set equal to 1 by the appropriate choice of the weight w. The same w is then used for measuring IRC of all grammars in $\Gamma_{k,n}$.

To illustrate this, let us consider G_1, G_E, and G_5, which all belong to $\Gamma_{2,3}$. As discussed above, G_1 is an optimal grammar in $\Gamma_{2,3}$ and $\text{IRC}_1 = 1$ by definition. Setting the right-hand side of (17) equal to 1, we get

$$w \cdot \frac{6}{3-0} \cdot \frac{2}{3} = 1,$$

which gives $w = 3/4$. The same value of w is used in (17) for all other grammars in $\Gamma_{2,3}$. Thus,

$$\text{IRC}_\text{E} = \frac{3}{4} \cdot \frac{6}{1-0} \cdot \frac{2}{3} = 3 \quad \text{and} \quad \text{IRC}_5 = \frac{3}{4} \cdot \frac{6}{6-4} \cdot \frac{2}{3} = \frac{3}{2}.$$

In general, when finding an optimal grammar within $\Gamma_{k,n}$ all possible matrices Φ should be considered and for each, ρ' and the maximum number

ρ^* of orders in R should be found. The greatest of the resulting γ values identifies an optimal grammar. The examples thus far are relatively simple and there are not too many possibilities to explore. In $\Gamma_{2,3}$, Φ has to look like in (10) and then it is easy to see that $\rho' = 6$ and $\rho^* = 3$. In $\Gamma_{3,3}$, the only choice of Φ is like in (11) and $\rho' = \rho^* = 6$. This is why G_2 is an optimal grammar in $\Gamma_{3,3}$. Then, since by definition $IRC_2 = 1$, (17) implies that $w = 1$ for the whole class:

$$w \cdot \frac{6}{6} \cdot \frac{3}{3} = 1 \Rightarrow w = 1 .$$

G_3 is in the same class of grammars and therefore

$$IRC_3 = 1 \cdot \frac{12}{6} \cdot \frac{3}{3} = 2 .$$

As illustrated by class $\Gamma_{3,3}$, it follows that $w = 1$ anytime $k = n$. This value is then extended to the case $k > n$ in which there is no optimal grammar. Grammar G_4 is in $\Gamma_{5,3}$ and $w = 1$ is used in (17) to evaluate its IRC:

$$IRC_4 = 1 \cdot \frac{24}{24} \cdot \frac{5}{3} = \frac{5}{3} .$$

Another $\Gamma_{5,3}$-grammar is considered below to illustrate further how ρ' is calculated.

Example 6. Let in this abstract example Φ be like in example 4, but let R have the following $\rho = 12$ elements:

$$R = Per\{(Nom_1, S), (Acc_1, O), P\} \cup Per\{(Nom_2, S), (Acc_2, O), P\} .$$

Suppose the parsing analysis, used to determine ρ', has access to set R. In this example, R shows also that some combinations of SSPFs are not permitted, e.g. S conveyed by Nom_1 cannot be combined with O conveyed by Acc_2. If this information is available to the parser, then $\rho' = 12$, making this grammar, G_6, equally efficient as G_4. This cannot be accepted since the restricted combinations of SSPFs represent additional rules in R and, therefore, G_6 should be more complex, i.e. ρ' should be greater than 12. For

this reason, the parser should only rely on Φ and *F*, and not at all on *R*. In this example, Φ shows that there are additional combinations of conveyors (not just $\mathrm{Nom}_1\mathrm{Acc}_1\mathrm{V}$, $\mathrm{Nom}_2\mathrm{Acc}_2\mathrm{V}$, and their permutations) that provide the information contained in set *F*. Whenever this happens, ρ' should be evaluated based on an enlarged set *R*, which contains *all* possible combinations of SSPFs, regardless of how they are conveyed. Such an enlargement makes the present *R* equal to the set in example 4. Therefore, $\rho' = 24$ and

$$\mathrm{AC}_6 = \frac{24}{12} \cdot 5 = 10\,, \qquad \mathrm{IRC}_6 = 1 \cdot \frac{24}{12} \cdot \frac{5}{3} = \frac{10}{3}\,.$$

Table 1 summarizes all AC and IRC values calculated up to this point. It shows how the scaling used to evaluate IRC changes AC to IRC. The relative position of each grammar within its class remains unchanged, but the IRC values are smaller and closer. IRC makes the grammars in $\Gamma_{2,3}$ comparable to $\Gamma_{3,3}$. This is like using the same yardstick to measure grammars that differ considerably, which shows one possible way of overcoming, on a small scale at least, what Miestamo (2006a, 2006b, to appear) calls the problem of comparability.

Class	Grammar	AC	IRC
$\Gamma_{2,3}$	G_1	4	1
	G_E	12	3
	G_5	6	1.5
$\Gamma_{3,3}$	G_2	3	1
	G_3	6	2
$\Gamma_{5,3}$	G_4	5	1.67
	G_6	10	3.33

Table 1. Complexity values for grammars in examples 1–6.

The proposed formula (17) is certainly not the only one that can be used for measuring grammar complexity in the present framework. Indeed, this formula has evolved from different versions that I have used in my work, and, even here, some modifications are proposed. However, my intention

from the very beginning (Vulanović 1991) has been to represent grammar efficiency as machine efficiency, which I have already mentioned in the introduction. In grammars, the useful output is the information that can be deduced from each sentence and the input consists of the grammatical devices that are used to convey this information. Therefore, the measure of grammar efficiency, *Eff*, can be defined as

$$(18)\quad Eff = \kappa \frac{Info}{Con},$$

where *Info* and *Con* are some appropriate measures of the information conveyed and of the conveyors respectively, and where κ is a constant of proportionality. Since I view grammar efficiency and complexity as reciprocal to each other, (18) is nothing else but the reciprocal of the IRC formula (17):

$$(19)\quad Eff = \mathrm{IRC}^{-1} = \frac{1}{w} \cdot \frac{n}{k\gamma},$$

with $\kappa = 1/w$, $Info = n$, and $Con = k\gamma$. *Maximally efficient grammars* in Vulanović (2003) are what I call here "optimal grammars." The process of transforming the input into the output, as modeled by mapping Φ and set *R*, is also represented in *Con* through the γ factor. At an earlier stage (Vulanović 1993), *Con* was represented as $k + \gamma'$, with γ' denoting a weighted version of γ. The switch to (19) was made because of simplicity: *w* is the only weight needed in this formula. In the future, a need may arise to fine-tune (19) further and include some additional weights in it. It is not clear at this stage how the new weights should be defined. I do not have enough intuition to tell me how to compare complexities of G_E, G_5, and G_3 for instance. So, until there is an indication that new weights are needed, it seems reasonable to keep them as simple as possible, i.e. all of them, except *w* in some cases, equal to 1. Moreover, there are not so many weight-assigning possibilities as it may seem. All of them reduce to the following two: redefine *w* and introduce a weight for ρ. If some weights are given to *n*, *k*, or ρ', they, together with $1/w$, form a new coefficient of $n/(k\gamma)$. This is equivalent to redefining *w*. Also, if a weight *a* is assigned to ρ_0, it can be factored out of the expression $\rho - a\rho_0$, which then changes *w* and the coefficient of ρ.

I am not sure how to interpret Hawkins' statement (2) and his stand on the relation between complexity and efficiency. How does efficiency "involve more or less complexity?" What is the "required minimum of complexity?" The latter may be related to my optimal grammars, but I could not find in Hawkins (2004) a definite explanation for (2). I only can comment on Hawkins' concept of grammar efficiency. It is obviously very different from mine, since I do not use phrase-structure trees, nor do I speak in terms of preferences of the human processor, which I have no intention of emulating. There are no relations of combination and dependency in my model, so there is nothing in it like Hawkins' principle (3i). However, my model seems to have some common points with principles (3ii) and (3iii) when the stated preferences are understood as descriptions of factors that increase efficiency. If *k* is interpreted as the "number of forms" of (3ii), then there is an agreement between *Eff* in (19) and (3ii) in the sense that *Eff* increases when *k* becomes smaller. Similarly, if *n* is viewed as the "properties that are assignable" of (3iii), then there is a connection between (3iii) and (19) because greater values of *n* increase *Eff*. On the other hand, there seem to be deeper differences between my concept of IRC, as given in (17), and language complexity as seen by Hawkins. While I separate forms and their syntactic/semantic/pragmatic properties as, respectively, input and output in my model, Hawkins considers them factors equally contributing to complexity, see (1). This indeed looks to me like a double count—a language has many forms mainly because they are needed to mark many SSPFs. In (17), complexity increases only if there is an unnecessary form, that is, if the same linguistic information can be conveyed with fewer forms without increasing word order restrictions. McWhorter's criterion (4iii) (and (5i) to some extent) also lists semantic and pragmatic distinctions as complexifying factors. Such distinctions are represented in (17) by *n* and, therefore, their increasing number diminishes(!) complexity, everything else being equal. This is because SSPFs are viewed as the output, something we get from the grammar. In reality, though, greater complexity can be expected when *n* is increased, since greater values of *n* are typically accompanied with greater *k* and ρ', while, usually, ρ is not maximized.

5. Further examples

In Vulanović (2003), n, as a factor in (19), represents a more complicated system (denoted by $\mathcal{F}$) of SSPFs. In the previous examples, $\mathcal{F}$ is simply $\{F\}$, but it may be a family of several sets of SSPFs. Examples in this section illustrate what is meant by $\mathcal{F}$. There is no ambiguity in any of them, thus $\rho_0 = 0$. Also, $k \geq n$ so that $w = 1$ and the connection between AC and IRC is simply AC = n·IRC. For this reason, only IRC values are calculated below.

Example 7. Consider simple intransitive and transitive sentences in the absence of object marking. There are three conveyors: N, intransitive verbs Vi, and transitive verbs Vt. An intransitive sentence conveys S and P, whereas a transitive sentence conveys S, O, and P, which means that $\mathcal{F}$ = {{S, P}, {S, O, P}}. The corresponding mapping is

$$\Phi: \quad \mathrm{N} \rightarrow \mathrm{S, O}, \quad \mathrm{Vi} \rightarrow \mathrm{P}, \quad \mathrm{Vt} \rightarrow \mathrm{P},$$

and $k = n = 3$. Suppose word order is rigid,

$$R = \{\mathrm{S(Vi, P)}, \mathrm{S(Vt, P)O}\},$$

thus $\rho = 2$. It also holds that $\rho' = 6 + 4 = 10$, a count resulting from 6 attempted parses of transitive sentences (like in example 1) and 4 attempted parses of intransitive sentences: two of them are SP and PS (recall from example 6 that all possible orders should be taken into account, not just those permitted in R), while OP and PO are the other two—they are attempted before it is realized that there is no other N in the sentence (the information that Vi only requires one N is stored in R and is not used in parsing). The above counts give

$$\mathrm{IRC} = 1 \cdot \frac{10}{2} \cdot \frac{3}{3} = 5 .$$

If the number of orders in R is increased to $\rho^* = 2 + 3 = 5$, which still preserves all sentences unambiguous, then IRC decreases to 10/5 = 2.

Example 8. In order to bring the above example closer to English, let us now assume that there is a class of verbs, denoted simply as V, which can be used both transitively and intransitively. This structure is modeled as

$$\Phi: \quad N \rightarrow S, O, \quad Vi \rightarrow P, \quad Vt \rightarrow P, \quad V \rightarrow P,$$

and

$$R = \{S(Vi, P), S(Vt, P)O, S(V, P), S(V, P)O\}.$$

It should be intuitively clear that this grammar is more complex than the one in example 7. The IRC measure confirms this: $k = 4$, $n = 3$, $\rho = 4$, and $\rho' = 6 + 6 + 4 + 4 = 20$, giving

$$\text{IRC} = 1 \cdot \frac{20}{4} \cdot \frac{4}{3} = \frac{20}{3} = 6.67 .$$

The numbers contributing to the value of ρ' are: 6 attempted parses of the three permutations of NNVt; 6 attempted parses of the three permutations of NNV; 4 attempted parses of the two permutations of NVi; and 4 attempted parses of the two permutations of NV.

In this case, ρ can be increased to $\rho^* = 2 + 3 + 2 + 3 = 10$, which reduces IRC to 8/3 = 2.67. This grammatical structure is still more complex than the corresponding one in example 7.

Example 9. Luiseño, a Uto-Aztecan language (Steele 1978), has free word order and makes a difference between animate (An) and inanimate (In) nouns. The unmarked form of these nouns is used to indicate S in the case of animate nouns and O in the case of inanimate nouns, which cannot be used as subjects. Animate nouns have also a marked form, denoted here as An-Acc, to indicate O. Simple transitive sentences with this structure can be modeled by the mapping

$$\Phi: \quad An \rightarrow S, \quad An\text{-}Acc \rightarrow O, \quad In \rightarrow O, \quad V \rightarrow P,$$

and by

$$R = Per\{S, (An\text{-}Acc, O), P\} \cup Per\{S, (In, O), P\}.$$

In this case, $k = 4$, $n = 3$, and $\rho = \rho' = 3! + 3! = 12$, which implies

$$\text{IRC} = 1 \cdot \frac{12}{12} \cdot \frac{4}{3} = \frac{4}{3} = 1.33 .$$

IRC > 1 since this is not an optimal grammar.

In the next three examples, simple active and passive transitive sentences are modeled together, forming a different family $\mathcal{F}$ from the one in examples 7 and 8. In addition to S, O, and P, agent (in the sense that English passives have agents), denoted here by A, is another SSPF to be conveyed. This means that $n = 4$. The two types of sentences correspond to $\mathcal{F}$ ={{S, O, P}, {S, A ,P}}. In all three grammars, A is marked by the agentive (Agt) case.

Example 10. This is an abstract example, used for a comparison to examples 11 and 12 below. It shows that one verbal form suffices for conveying $\mathcal{F}$. Let

Φ: Nom → S, Acc → O, Agt → A, V → P,

and

R = {PSO, PAS}.

It holds that $k = 4$, $\rho = 2$, and $\rho' = 3! + 3! = 12$, implying

$$\text{IRC} = 1 \cdot \frac{12}{2} \cdot \frac{4}{4} = 6 .$$

When ρ is increased to $\rho^* = 3! + 3! = 12$, we have an optimal grammar because IRC = 1.

Example 11. Consider the structure like in English, where the phrase "by N" is meant as Agt,

Φ: Nom → S, O, Agt → A, Act → P, Pas → P.

There are $\rho = 2$ strings in R,

R = {S(Act, P)O, S(Pas, P)A}.

Although here $k = 4$, like in the previous example, the conveyors are used in a more complicated way. In particular, two verbal forms are not necessary. Even if all 3 permutations of S(Act, P)O and all 6 permutations of S(Pas,P)A are included in R (this still preserves unambiguity of all sentences), the grammar remains far from an optimal one. This is because there are other conveyor combinations that produce the information in $\mathcal{F}$. Based on the discussion in example 6, all those combinations should be considered when ρ' is calculated. Hence, the count of parsing attempts is applied to the sentences Nom V Nom and Nom V Agt and all their permutations, where V stands for both Act and Pas. This gives $2(3 + 6) = 18$ sentences in all, with a combined number of $\rho' = 2(6 + 6 + 3) = 30$ parsing attempts. Each of the three permutations of Nom V Nom has two parses (cf. example 1); the three permutations of Nom V Agt, in which Nom precedes Agt, also require two parsing attempts each (since Nom can initially be interpreted as either S or O); and finally, there is one parse of each of the three permutations of Nom V Agt in which Agt precedes Nom (Agt is unambiguously interpreted as A and then Nom has to be S). Formula (17) implies

$$\mathrm{IRC} = 1 \cdot \frac{30}{2} \cdot \frac{4}{4} = 15,$$

which indicates greater complexity than in example 10 or in the case of active sentences alone (recall that these are modeled by G_E, for which IRC = 3).

If all possible unambiguous word orders are permitted, ρ increases to $\rho^* = 2(3 + 6) = 18$. This decreases IRC to $30/18 = 5/3 = 1.67$.

Example 12. The structure of Maori (Hohepa 1969, Chung 1978, Vulanović 1997) is similar to the grammar in example 10, but, like English, Maori has active and passive verbal forms. Therefore,

Φ: Nom → S, Acc → O, Agt → A, Act → P, Pas → P.

Fixed word order,

R = {(Act, P)SO, (Pas, P)AS},

is assumed for simplicity, and this enables a direct comparison to examples 10 and 11. Regardless of R, 24 sentences should be considered in order to find ρ'. This is so because there are 6 permutations of each Nom Acc V and Nom Agt V, where V = Act, Pas. Each of the permutations has exactly one parse, thus $\rho' = 24$. Since $k = 5$ and $\rho = 2$, it follows that

$$\mathrm{IRC} = 1 \cdot \frac{24}{2} \cdot \frac{5}{4} = 15,$$

which happens to be the same as in example 11. The English model has one conveyor less, but greater processing difficulty. However, if in the Maori model ρ is maximized to $\rho^* = 24$, which still leaves all sentences unambiguous, then the above IRC drops to 5/4 = 1.25. This is less complex than the corresponding structure with maximized ρ in example 11.

Example 13. This last example describes formally a situation similar to McWhorter's (2001) comparison of Kikongo and Japanese (see the introduction).

Consider three languages which have almost identical grammatical structures, the only difference being that languages A and B have one more verbal form than language C. The extra verbal form is used in A to convey a tense which does not exist in C, whereas in B it merely duplicates the usage of another verbal form. To represent this, it is sufficient to model simple intransitive sentences (transitive sentences or intransitive and transitive sentences modeled together give similar results). Suppose word order is free and there are no ambiguous sentences, so that $\rho = \rho'$ and $\rho_0 = 0$. This gives $\gamma = 1$ in (16) and (17). Let languages A and B have $k = m + 1$ conveyors: N and m verbal forms. Each of the verbal forms conveys a different tense in A, whereas B has only $m - 1$ tenses. Therefore, the system of SSPFs is

$$\mathcal{F} = \{\{\mathrm{S}, \mathrm{P}_i\} \mid i = 1, 2,..., m\},$$

where the tenses are denoted by P_i, $i = 1, 2,..., m$ ($\mathrm{P}_{m-1} = \mathrm{P}_m$ in B). For language A, $n = m + 1$, so that $\mathrm{IRC_A} = 1$. The grammar of A is therefore optimal. At the same time, $n = m$ in B and $\mathrm{IRC_B} = (m + 1)/m > 1$.

Consider now language C with $k = m$ conveyors: N and $m - 1$ verbal forms, each of which conveys a different tense. Since $n = m$, it follows that $IRC_C = 1$.

Therefore, A and C have the same information-relative complexity and B is more complex. Absolute complexity does not reveal the difference between A and B. Formula (16) gives $AC_A = AC_B = m + 1$, which indicates greater complexity than in C, where $AC_C = m$.

6. Conclusion

In this paper, I presented a model-based mathematical formula for measuring grammar complexity. The formula is a modification of the metric that I proposed in Vulanović (2003). Through a detailed derivation process, illustrated by many examples, I explained the formula and compared it to complexity criteria by other authors. Although the final formula is by no means unique, I was guided by the concept of machine efficiency (complexity is defined as reciprocal to efficiency) that I extended to grammars for the first time in Vulanović (1991). When there were choices how to put complexity factors together, I followed what seemed to be the simplest option. It should be pointed out that in this model there is no separate definition of grammar complexity followed by a formula for measuring it. In fact, the formula *is* the definition of grammar complexity in the model. This is why the only way of confirming the validity of the formula is to see what kind of numerical values it assigns to the complexity of grammatical structures that can be ranked based on intuition. Usually, when we compare two structures that are relatively similar, we can tell which one is more complex. In such cases, as illustrated by several examples, the complexity formula correctly assigns a greater value to the more complex grammar. This can be used to represent various syntactic changes, since a syntactic change can be viewed as a sequence of slightly different grammars. Then the complexity (or efficiency) of each grammar can be measured using the proposed formula, see Vulanović (1997, 2005a) for instance.

Other questions about how "realistic" the formula is are hard to answer because there is no ground for comparison. For instance, we have found complexities of two grammars, G_E and G_2, to be 3 and 1.5 respectively. Does this mean that G_E is exactly twice as complex as G_2? Yes, in the model! However, in order to answer this outside the model, we have to know what "exactly twice as complex" means. At this stage, there is no other definition or measure of complexity that can tell us this.

The formula measures what I call the *information-relative complexity*. Example 13 shows that this kind of complexity gives more reliable results than *absolute complexity*, the other type of complexity considered in this paper. Three hypothetical languages, A, B, and C, are compared in example 13. The only difference between them is in the number of verbal forms and tenses. A and B have the same number of verbal forms and C has one verbal form less. Because of this, languages A and B have the same absolute complexity, while the absolute complexity of C is less. However, the need for an additional verbal form can be justified in A by the need to convey one more tense, but this justification cannot be extended to B, which has the same number of tenses as C. Thus, relative to the information (tenses in this case) that the languages have to convey, A is less complex than B. This shows that absolute complexity is not enough for an accurate comparison of different grammatical structures. It can be used to compare languages that convey the same type and amount of linguistic information, otherwise, comparisons based on absolute complexity may be misleading. Information-relative complexity places such languages in a correct relationship.

If it seems reasonable to compare some languages, like A and B above, on the scale of information-relative complexity, then why should not all languages be compared on the same scale? When A and C are compared like that, it turns out that they are equally complex. For information-relative complexity, the main question is how complex the language is in comparison to the simplest possible structure that conveys the same amount of linguistic information. This is what the present paper is mainly about.

References

Chung, Sandra (1978) *Case Marking & Grammatical Relations in Polynesian*. Austin: University of Texas Press.

Dahl, Östen (2004) *The Growth and Maintenance of Linguistic Complexity*. Amsterdam/ Philadelphia: Benjamins.

Dik, Simon C. (1997) *The Theory of Functional Grammar. Part 1: The Structure of the Main Clause*. Berlin: Mouton de Gruyter.

Frazier, Lyn (1985) Syntactic complexity. In David R. Dowty, Lauri Karttunen & Arnold M. Zwicky (eds.), *Natural Language Parsing: Psychological, Computational, and Theoretical Perspectives*, pp. 129–189. Cambridge: Cambridge University Press.

Gil, David (2001) Creoles, complexity, and Riau Indonesian. *Linguistic Typology* 5: 325–371.

Hawkins, John A. (1994) *A Performance Theory of Order and Constituency*. Cambridge: Cambridge University Press.

—— (2004) *Efficiency and Complexity in Grammars*. Oxford/New York: Oxford University Press.

Hengeveld, Kees, Rijkhoff, Jan & Siewierska, Anna. (2004) Parts-of-speech systems and word order. *Journal of Linguistics* 40: 527–570.

Hohepa, Patrick (1969) The accusative-to-ergative drift in Polynesian languages. *Journal of Polynesian Society* 78: 295–329.

Juola, Patrick (1998) Measuring linguistic complexity: The morphological tier. *Journal of Quantitative Linguistics* 5: 206–213.

Kusters, Wouter (2003) *Linguistic Complexity, the Influence of Social Change on Verbal Inflection*. [Ph.D. Diss., University of Leiden]. Utrecht: LOT. [Cited in (Miestamo 2006a, 2006b, to appear).]

McWhorter, John H. (2001) The world's simplest grammars are creole grammars. *Linguistic Typology* 5: 125–166.

—— (to appear a) Why does a language undress? Strange cases in Indonesia. In Matti Miestamo, Kaius Sinnemäki & Fred Karlsson (eds.), Amsterdam: Benjamins.

—— (to appear b) *Language Interrupted: Signs of Non-native Acquisition in Standard Language Grammars*. New York: Oxford University Press. [Cited in (McWhorter, to appear a).]

Miestamo, Matti (2006a) On the feasibility of complexity metrics. In Krista Kerge & Maria-Maren Sepper (eds.), *FinEst Linguistics, Proceedings of the Annual Finnish and Estonian Conference of Linguistics*, *Tallin, May 6–7, 2004*. Publications of the Department of Estonian of Tallinn University 8, pp. 11–26, Tallinn: Tallin University Press.

—— (2006b) On the complexity of standard negation. In Mickael Suominen, Antti Arppe, Anu Airola, Orvokki Heinämäki, Matti Miestamo, Urho Määttä, Jussi Niemi, Kari K. Pitkänen & Kaius Sinnemäki (eds.), *A Man of Measure: Festschrift in Honour of Fred Karlsson on his 60th Birthday. Special Supplement to SKY Journal of Linguistics*, Vol. 19, pp. 345–356.

—— (to appear) Grammatical complexity in a cross-linguistic perspective. In Matti Miestamo, Kaius Sinnemäki & Fred Karlsson (eds.), Amsterdam: Benjamins.

Miestamo, Matti, Sinnemäki, Kaius, and Karlsson, Fred (eds.) (to appear) *Language Complexity: Typology, Contact, Change. Studies in Language Companion Series*. Amsterdam: Benjamins.

Miller, George A., and Chomsky, Noam (1963) Finitary models of language users. In R. Duncan Luce, Robert Bush & Eugene Galanter (eds.), *Handbook of Mathematical Psychology*, Vol. 2, pp. 419–492, New York: Wiley.

Nichols, Johanna (1992) *Linguistic Diversity in Space and Time*. Chicago/London: The University of Chicago Press.

Steele, Susan (1978) Word order variation: A typological study. In Joseph H. Greenberg (ed.), *Universals of Human Language*, pp. 587–623, Stanford: Stanford University Press.

Vulanović, Relja (1991) On measuring grammar efficiency and redundancy. *Linguistic Analysis* 21: 201–211.
——— (1993) Word order and grammar efficiency. *Theoretical Linguistics* 19: 201–222.
——— (1997) Model-based measuring of syntactic change. *Journal of Quantitative Linguistics* 2: 67–76.
——— (2003) Grammar efficiency and complexity. *Grammars* 6: 127–144.
——— (2005a) The rise and fall of periphrastic *do* in affirmative declaratives: A grammar efficiency model. *Journal of Quantitative Linguistics* 12: 1–28.
——— (2005b) The combinatorics of cases and word order. *Research on Language and Computation* 3: 107–129.
Wolfart, H. Christoph & Carroll, Janet F. (1981) *Meet Cree: A Guide to the Cree Language*. Lincoln: University of Nebraska Press.

Contact information:

Relja Vulanović
Department of Mathematical Sciences
Kent State University Stark Campus
6000 Frank Ave. NW
North Canton, Ohio 44720
USA
e-mail: rvulanov(at)kent(dot)edu