An insertion into the Chomsky hierarchy?

Robert McNaughton
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180-3590, U.S.A.
mcnaught@cs.rpi.edu

January, 1999

Abstract. This review paper will report on some recent discoveries in the area of Formal Languages, chiefly by F. Otto, G. Buntrock and G. Niemann. These discoveries have pointed out certain break-throughs connected with the concept of growing context-sensitive languages, which originated in the 1980's with a paper by E. Dahlhaus and M.K. Warmuth. One important result is that the deterministic growing context-sensitive languages turn out to be identical to an interesting family of formal languages definable in a certain way by confluent reduction systems.

1. Growing context-sensitive languages.

There are several reasons for proposing that the family of GrCSL's (growing context-sensitive languages) be considered as a new level in the Chomsky hierarchy of languages. The insertion would be between the CFL's (context-free languages) and the CSL's (context-sensitive languages), in effect making a new family designated as the "type one-and-a-half languages". This family began to receive attention in 1986, when Dahlhaus and Warmuth published their result [5] that the complexity of its membership problem had a polynomial-time algorithm, in contrast to the P-space-complete membership problem for the larger family of context-sensitive languages.

A GrCSG (growing context-sensitive grammar) is one in which $|\alpha| < |\beta|$ for every rule $\alpha \to \beta$; a GrCSL is the language of a GrCSG. This class of grammars is a proper subclass of the class of CSG's (context-sensitive grammars) where rules are also permitted in which $|\alpha| = |\beta|$.

The GrCSL's are just one of many studied families properly between the CFL's and the CSL's. Another was the family of CSL's with linear-bounded derivations (see, e.g.,

Ron Book's dissertation [2]), i.e., languages having a CSG with a bound B such that every $w \in \Sigma^*$ derived in the grammar has a derivation whose length is $\leq B|w|$. In 1964 Gladkij proved in [1] that the CSL

$$\{wcw^Rcw | w \in \{a,b\}^*\}$$

does not have linear-bounded derivations (see also the appendix to [2]). (w^R means w written backwards.)

More recently it has been proved that the GrCSL's are a proper subfamily of the family of CSL's with linear bounded derivations (see [12], and also [13], Corollary 5.4). However, the latter family will no longer be of concern in this paper.

From the work of Lautemann [6] and Buntrock [12] it follows that the language $\{ww | w \in \{a,b\}^*\}$ is not a GrCSL. (See also [13], especially the penultimate paragraph of Section 1.) Thus we have an improvement on the Gladkij language for a paradigm CSL that is not a GrCSL.

All CFL's consisting of words of length ≥ 2 are GrCSL's. The easy proof is by a constructive modification of the Chomsky normal form. But not all GrCSL's are context-free, e.g, $\{ba^{2^n} | n \geq 1\}$, a GrCSG for which is

$$S \to SK | baa$$

$$baaK \to baaaa$$

$$aK \to Kaa$$

The concept GrCSG is based on the length of strings. It is convenient to allow as GrCSG's the grammars that satisfy a variant of the definition based on the weighted length of strings. A *weighting function* ϕ on the words over an alphabet Σ maps each word to an integer satisfying the following: (1) $\phi(x) > 0$ for all $x \in \Sigma$, (2) $\phi(\lambda) = 0$ (λ is the null string) and (3) $\phi(xy) = \phi(x) + \phi(y)$ for all words x and y. We define a grammar to be a *GrCSG in the new sense* if there is a weighting function ϕ on the words over the total alphabet such that $\phi(\alpha) < \phi(\beta)$ for every rule $\alpha \to \beta$. Following [8] and [11] we can prove that, if G is a GrCSG in the new sense then there is a G' that is a GrCSG in the original sense such that $L(G') = L(G) \cap \Sigma\Sigma\Sigma^*$.

Note that a context-free grammar in Chomsky normal form whose language does not have the null word is a GrCSG in the new sense: take $\phi(x) = 1$ for x a variable and $\phi(x) = 2$ for x a terminal.

It will be convenient to adopt this new definition of GrCSG for the remainder of this paper, yielding the slight change in the definition of GrCSL.

As is well known, a language is context-sensitive if and only if it is recognized by a nondeterministic LBA (linear bounded automaton). We can get a corresponding

result for GrCSL's by modifying this automaton to one whose tape decreases in its weighted length at every move. The best way to work out this idea precisely is to follow Buntrock and Otto in Section 3 of [13], stipulating an automaton with two pushdown tapes, representing the portions of the LBA tape to the left of the head and the right of the head, respectively. An elaborate weighting function is defined for configurations of the automaton, satisfying the condition that if one configuration is followed by another then the weight of the latter is less than the weight of the former. This weighting function is based on a weighting function of words over the alphabet, but it is far too complicated to be described here. (One trick is to get the effect of weighing each character of Σ more heavily on one pushdown than on the other.)

We thus have a nondeterministic shrinking two-pushdown automaton and the result that a language is a GrCSL if and only if it is recognized by such a device. The proof given by Buntrock and Otto is somewhat similar to the proof that a language is a CSL if and only if it is accepted by a nondeterministic LBA.

The question naturally arises as to which CSL's are GrCSL's and which are not. No broad answer has been given to this question. Some well known CSG's using length-preserving rules have languages that turn out to be GrCSL's, an example being the grammar:

$$S \to SABC|dABC$$

$$BA \to AB \qquad CB \to BC \qquad CA \to AC$$

$$dA \to da \qquad aA \to aa \qquad aB \to ab$$

$$bB \to bb \qquad bC \to bc \qquad cC \to cc$$

Its language $\{da^nb^nc^n|n \geq 1\}$ also has the GrCSG:

$$S \to SK|SL|dabc$$

$$cK \to Kcc \qquad bK \to Kbb$$

$$aK \to Kaa \qquad daK \to daa$$

$$cL \to Mccc \qquad cM \to Mcc \qquad bM \to Nbbb$$

$$bN \to Nbb \qquad aN \to Kaaa \qquad daN \to daaa$$

The weighting function is $\phi(x) = 1$ if x is a variable, $\phi(x) = 2$ if x is a terminal. (To derive $da^{32}b^{32}c^{32}$ in this grammar we would begin by deriving $dabcK^5$. But to derive $da^{37}b^{37}c^{37}$ we would note that $37 = 2^5 + 2^2 + 2^0$, and accordingly we would begin by deriving $dabcKKLKL$.)

With a bit more trouble we could get a GrCSG for $\{da^nb^nc^n\}$ in the original sense, i.e., one in which the weighting function is $\phi(x) = 1$ for all variables and terminals x. Also, if we wished to get rid of the d, which acts as a left-end marker, we could do so at the expense of further complication in the grammar.

GrCSG's have the advantage over other CSG's in that, in each such grammar, the length of a derivation has an upper bound that is linear in the length of the word derived. Moreover, as mentioned, the membership problem for every GrCSL has a polynomial-time algorithm. (The proof in [5] goes over to our new definition of GrCSL.)

Dahlhaus and Warmuth [5] prove that every GrCSL is log-tape reducible to some CFL. Buntrock and Otto [13] improve on this result by showing that this reduction can be done as a one-way log-space reduction; that is to say, the GrCSL is (quoting Section 1 of [13]) "accepted by an auxiliary pushdown automaton with logarithmic space bound and polynomial time bound that uses its input tape in a one-way fashion."

As pointed out in [8], the family of GrCSL's is an abstract family of languages, that is to say, this family is closed under union, concatenation, star iteration, intersection with regular languages, null-word-free homomorphisms and inverse homomorphisms.

A persistent problem for theoretical computer scientists has been to find a family of formal languages that will include all programming languages or most programming languages. The family must be reasonably simple conceptually and must not be so broad as to include languages that have properties that no reasonable programming language could have. This objective is necessarily vague and this is no place to attempt to refine it or even to discuss it, except to make a brief negative point about the family of GrCSL's: If the family of formal languages must include a language of programs whose variables occurring in executable statements must also occur in declaration statements, and if the family includes variables of unlimited length, then the family of GrCSL's is not a suitable family for this purpose. The argument for this negative assertion is based on the piece of evidence that $\{ww | w \in \{a, b\}^*\}$ is not a GrCSL, which indicates that any programming language in which a defined program may have arbitrarily long variables, but only those that are declared, is probably not a GrCSL.

The application of ideas from Theoretical Computer Science to actual computing is difficult to predict with any precision. But it helps to have some general idea of the possibility of some application, even if the exact nature of this application is vague. There will be more to say about the applicability of these ideas on GrCSL's when we investigate the deterministic variety of them in the next section.

2. The deterministic variety.

A GrCSL is *deterministic* if it is recognized by a deterministic shrinking two-pushdown automaton. The name for this automaton is rather long; let us call it a "D-shrink" for short. As it is an important concept, it deserves a formal definition. The following is adapted from the paper by Buntrock and Otto [13]:

A *D-shrink* is a 7-tuple

$$(Q, \Sigma, , , , \delta, q_0, B, F).$$

Here Q is the set of states, Σ the input alphabet, , the tape alphabet ($\Sigma \subset$, and , $\cap Q = \emptyset$), q_0 is the initial state, B is the bottom marker of the two pushdown stores ($B \in$, $- \Sigma$), F is the set of accepting states, and δ is the transition function:

$$\delta : Q \times , ^2 \rightarrow Q \times , ^* \times , ^* \cup \{\emptyset\}.$$

A configuration in a computation of a D-shrink is given as uqv where $u, v \in , ^*$ and $q \in Q$. The idea is that u and v are the words on the left and right pushdown tapes, respectively. Normally B is the leftmost character of u and the rightmost character of v and occurs nowhere else. Where $u = u'a$, $v = bv'$, $a, b \in$, and $\delta(q, a, b) = (q', w_1, w_2)$, the word $u'w_1q'w_2v'$ is the next configuration in that computation. If $\delta(q, a, b) = \emptyset$ then uqv is a halting configuration. An initial configuration is of the form Bq_0vB where v is the input. The bottom marker B is never created or destroyed in a computation but may be sensed. Acceptance of v is either by final state (a configuration $u'qv'$ where $q \in F$) or by empty store (a configuration BqB for $q \in Q$). The *language* of a D-shrink is the set of all accepted inputs.

What makes the D-shrink shrinking is the stipulation that there is a weighting function ϕ on strings over the alphabet $Q \cup$, such that, for $\delta(q, a, b) = (q, w_1, w_2)$, the condition $\phi(w_1qw_2) < \phi(aqb)$ holds.

As mentioned in Section 1, a language is a GrCSL if and only if it is the language of a nondeterministic shrinking two-pushdown automaton ([13], Section 3). If a GrCSL is accepted by a D-shrink (i.e., a deterministic automaton of the variety) then the language is said to be a *DGrCSL (deterministic growing context-sensitive language).* I shall argue in the remainder of this paper that the family of DGrCSL's is an important family of languages, perhaps more important than the larger family of all GrCSL's.

The language $\{ba^{2^n} | n \geq 1\}$ is a DGrCSL. One can design a D-shrink for it based on the grammar from Section 1:

$$S \rightarrow SK | baa$$

$$baaK \rightarrow baaaa$$

$$aK \rightarrow Kaa$$

Any word in the language of this grammar will be processed by the D-shrink according to a rightmost derivation in the grammar, which means that the word is processed from left to right. For example, the rightmost derivation of the word ba^{32} has the line ba^3KaaKK, to which the rule $aK \rightarrow Kaa$ is applied to the rightmost aK, resulting in the line $ba^3KaKaaK$. The automaton does things in reverse of the order in the derivation; for that step it might have Bba^3Ka on the left tape and KB on the right tape, and might be in a state showing that Kaa is between the ba^3Ka and the K. It then would push aK onto the left tape, and would go into a state showing that the null word is between the Bba^3KaaK on the left tape and the KB on the right tape.

We shall not verify that this automaton can be made deterministic, and therefore that the language $\{ba^{2^n}|n \geq 1\}$ is a DGrCSL. The language $\{da^nb^nc^n|n \geq 1\}$ is also a DGrCSL; details beyond the discussion of this language in Section 1 are omitted.

It is not difficult to prove that the family of DGrCSL's is closed under complementation. This observation enables us to prove the existence of CFL's that are not DGrCSL's. Such a language is $\{a,b\}^* - \{ww|w \in \{a,b\}^*\}$. If this CFL were a DGrCSL then its complement $\{ww|w \in \{a,b\}^*\}$ would also be a DGrCSL, and hence would be a GrCSL, which (as mentioned in Section 1) it is not.

And so, although all null-word-free CFL's are GrCSL's, they are not all DGrCSL's. This may be an unpleasantness that might dissuade some theoreticians from accepting the family of GrCSL's as a member of the Chomsky hierarchy. As it now stands each family in the hierarchy is a subclass of the deterministic subclass of the family at the next level. (Incidentally, as noted in [12] and [13], all null-word-free deterministic CFL's are DGrCSL's.)

Whether or not it deserves a place in the Chomsky hierarchy, the family of GrCSL's is an important family. Indeed there is reason to regard the subclass of DGrCSL's as being more important than the larger family of GrCSL's. As will be shown in the next section, the DGrCSL's can be characterized in terms of confluent rewriting systems, which gives them perhaps even more significance.

(Before going on to Section 3, let us pause to observe that the family of DGrCSL's is not closed under union or intersection. The simple argument for intersection [14] is as follows: It is easy to see that the Gladkij language $\{wcw^Rcw|w \in \{a,b\}^*\}$ is equal to the intersection of two deterministic CFL's, which are therefore both DGrCSL's; but the Gladkij language itself is not a DGrCSL. That this family is also not closed under union follows by the DeMorgan law, since the family is closed under complementation. Other such results can be found in Section 5 of [14].)

3. Confluent string rewriting systems.

Perhaps the greatest selling point for the family of GrCSL's is its link with the theory of rewriting systems as it has been developing since 1970. More specifically, the selling point is for the family of DGrCSL's, which turns out to be identical to a family of languages definable in a certain way by confluent string rewriting systems, as discovered recently by Niemann and Otto [14].

Briefly, a *string rewriting system* is a semi-Thue system. We focus on systems in which the application of a rule to a word results in a simplification of the word: for example, it may be that $|\beta| < |\alpha|$ holds for each rule $\alpha \to \beta$. Such systems are often called *reduction systems,* since their purpose is to take a long word and gain some sort of understanding by reducing it to a shorter word. The length requirement is not a

strict requirement, but one necessary property of a reduction system is that there be no infinite derivations. Consequently every word can be reduced to an irreducible word (the Noetherian property).

A further property that is desirable for a reduction system is that no word can be reduced to two distinct irreducible words (the confluence property). In reducing a word according to a confluent reduction system, it is sometimes possible to start reducing in two distinct ways at the same point in the word. But then the two reduction sequences must eventually come together. (For a good exposition of string rewriting systems, see the first two chapters of [9].)

Some languages can be defined by confluent reduction systems; if the alphabet of the system is the same as the alphabet of the language then the language is a *congruential language,* i.e., the union of some of the congruence classes of a congruence relation over Σ^*. Unfortunately, most interesting formal languages are not congruential.

However, if we allow ourselves to supplement the alphabet of the reduction system to include some control characters along with the alphabet of the language, we get something that is more fruitful. The paper [7] investigated the question of which formal languages could be defined in this way. The results and conceptual development of that paper have recently been surpassed in a remarkable way by Niemann and Otto [14], whom the present exposition will follow.

Another desirable property that our reduction systems must have is that they be weight reducing in the sense defined in Section 1; that is to say, there is a weight function ϕ such that, for every rule $\alpha \to \beta$, $\phi(\beta) < \phi(\alpha)$. A reduction system has the *generalized Church-Rosser property* if it is confluent and weight reducing. (It has the *strict Church-Rosser property* if it is confluent and length-reducing. Except for a few isolated remarks, we shall generally ignore the strict property for the remainder of this paper in favor of the more general property.)

A language $L \subseteq \Sigma^*$ is a *GenCRL (generalized Church-Rosser language)* if there exists a reduction system S with the generalized Church-Rosser property satisfying the following conditions:

(1) The alphabet , of S contains Σ as a proper subset;
(2) There are Y, t_1 and t_2, where $t_1, t_2 \in (, -\Sigma)^*$, $Y \in , -\Sigma$ and Y is irreducible in S, such that, for all $w \in \Sigma^*$, $w \in L$ if and only if $t_1 w t_2 \to^* Y$ (viz., Y is derivable from $t_1 w t_2$ in S)

Notice that if we have a language L that is of interest to us then such a system S would be a nice thing to have for testing membership in L. Given any $w \in \Sigma^*$, we would form the word $t_1 w t_2$ and reduce it modulo S. If the reduced word is Y then $w \in L$; if not then it is not. Since every rule of S is weight-reducing, the length of the reduction of $t_1 w t_2$ is linear in $|\phi(w)|$, and hence linear in $|w|$. Each step of the reduction is rather

easy; we simply scan the word that we have for a subword that is the left side of a rule. (Things can be done so that the total amount of time spent in scanning during the entire reduction is insignificant.) When we find such a subword we reduce the word accordingly. If we find that there is no such subword and the word at that point is not simply Y then we know that the original word w is not in L.

As proved in [13] and [14], a language L is a GenCRL if and only if it is a DGrCSL. In effect, the D-shrink is a suitable mechanism for reduction of the word in the reduction system; in fact, precisely suitable. This automaton is similar to the automaton conceived by Ron Book [4] to reduce a word according to a reduction system with the Church-Rosser property.

In [14] it is demonstrated that every GenCRL is also a CRL; in other words the reduction system can be modified so as to allow the length function as the weighting function (i.e., for each $x \in$, , $\phi(x) = 1$). In the same paper, it is also demonstrated that the reduction system can be modified so that, for every $w \in \Sigma^*$, $t_1 w t_2$ reduces either to Y (indicating "yes," that $w \in L$), or to N (indicating "no," that it is not); both $N$and Y are in , $-\Sigma$. These results settled questions left open in [7]. Furthermore, they make the characterization of DGrCSL's in terms of rewriting systems even more significant than they appear at first. They also strengthen our feeling that the DGrCSL's constitute an important family of languages.

An interesting open question concerns the language $\{ww^R | w \in \{a,b\}^*\}$, which is clearly a GrCSL, since it is context-free. I conjecture, however, that it is not a DGrCSL. In [7] there is a plausibility argument that it is not a CRL.

In conclusion, I suspect there will probably be few theoreticians who will press for any modification of the Chomsky hierarchy. Nevertheless, I hope many will come to realize that both the family of GrCSL's and the family of DGrCSL's will play important roles in the future of computer science. I am especially convinced of the importance of the DGrCSL's, since they have such a solid link with the contemporary theory of rewriting systems, and, in particular, with string rewriting systems having the confluence property.

References

[1] Gladkij, A.W., "On the complexity of derivations in context-sensitive grammars," *Algebri i Logika Sem.,* vol 3, pp. 29–44, 1964. In Russian.

[2] Book, R.V. *Grammars with time functions,* Dissertation, Harvard University, 1969.

[3] Salomaa, A., *Theory of Automata,* Pergamon Press, Oxford, England, 1969.

[4] Book, R.V., "Confluent and other types of Thue systems," *J. ACM,* vol. 29, pp. 171–182, 1982.

[5] Dahlhaus, E. and M.K. Warmuth, "Membership for growing context-sensitive grammars is polynomial," *J. Computer and System Science,* vol. 33, pp. 456–472, 1986.

[6] Lautemann, C. "One pushdown and a small tape," in *Dirk Siefkes zum 50. Geburtstag* (K.W. Wagner, ed.), pp. 42–47, Technische Universität Berlin and Universität Augsburg, 1988.

[7] McNaughton, R., P. Narendran and F. Otto, "Church-Rosser Thue systems and formal languages," *J. ACM,* vol. 35, pp. 324–344, 1988.

[8] Buntrock, G. and Loryś, K., "On growing context-sensitive languages," *Proc. 19th ICALP, Lecture Notes in Computer Science* (W. Kuich,ed.), vol. 623, pp. 77–88, 1992.

[9] Book, R.V. and F. Otto, *String-rewriting systems,* Springer-Verlag, 1993.

[10] Buntrock, G., "Growing context-sensitive languages and automata," Preprint-Reihe, Nr. 69, Inst. für Informatik, Universität Würzburg, 1993.

[11] Buntrock, G. and Loryś, K., "The variable membership problem: succinctness versus complexity," *Proc. 11th STACS, Lecture Notes in Computer Science* (P. Enjalbart, E.W. Mayr and K.W. Wagner, eds.), Springer, pp. 77–88, 1994.

[12] Buntrock, G., *Wachsend kontextsensitive Sprachen,* Habilitationsshrift, Fakultät für Mathematik und Informatik, Universität Würzburg. 1996.

[13] Buntrock, G. and F. Otto, "Growing context-sensitive languages and Church-Rosser languages," *Inf. and Computation,* vol. 141, pp. 1–36, 1998.

[14] Niemann, G. and F. Otto, "The Church-Rosser languages are the deterministic variants of the growing context-sensitive languages," *Proc. Foundations of software science and computation structures; Lecture notes in Computer Science,* vol. 1378, Springer-Verlag, pp. 243–257, 1998.