ISSN 0105-8517

SPACE-RESTRICTED ATTRIBUTE GRAMMARS

by

Erik Meineche Schmidt

DAIMI PB-117
April 1980

Computer Science Department
AARHUS UNIVERSITY
Ny Munkegade – DK 8000 Aarhus C – DENMARK
Telephone: 06 – 12 83 55

Space-restricted attribute grammars

Erik Meineche Schmidt
Computer Science Department
Aarhus University
DK-8000 Aarhus C, Denmark

Abstract

Restricting the size of attribute values, relative to the length of the string under consideration, leads to a model of attribute grammars in which grammars with both inherited and synthesized attributes can be significantly more economical than grammars with synthesized attributes only.

1. Introduction

When Knuth introduced the notion of an attribute grammar ([Knu]) as a formalization of the concept of assigning meaning to strings generated by context-free grammars, the definition allowed any collection of sets as attribute values and any collection of functions over these sets as semantic functions. This generosity has the immediate consequence that there is no real need for inherited attributes, because any translation defined by an arbitrary attribute grammar can be defined by another attribute grammar which uses only synthesized attributes. Although this is a correct observation, one has the feeling (as also pointed out in [Knu]) that it doesn't tell the whole story and that there are many situation - handling declarations in a programming language for example - where the use of inherited attributes is both natural and advantageous. One way of turning this feeling into mathematical results is to restrict the use of semantic domains in such a way that one cannot "drag along" the whole derivation tree as an attribute value, and then apply a function at the root which maps the tree into whatever translation is wanted (this is the argument that makes inherited attributes obsolete).

There are several papers where this approach has been taken - we know of [Dus], [EF], [LRS] and [Ri], where translations defined by attribute grammars over fixed domains are analyzed and compared. It is common to these approaches that the restrictions on the domains are "syntactic" in nature, exemplified by [EF], where the main concern is with domains whose values are strings or trees, and whose operations are string- or tree-concatenation. We shall also restrict attention to domains whose values are strings, but instead of restricting the semantic functions, we take a more information theoretic approach in which we bound the size of the attribute values relative to the length of the word under consideration.

Inspired by the definition of spacebounds in complexity theory, we call an attribute grammar S(n)-spacebounded if there exists a constant c,such that for any word w generated by the underlying context-free grammar, all attribute values in all derivation trees for w are of length at most c $S(|w|)$. We repeat that attribute values are strings.

It follows that if the spacebound is sublinear, then the "drag along the whole tree"-approach to elimination of inherited attributes can only be used at the cost of expanding the bound. The question then becomes whether there is another, less expensive,way of eliminating inherited attributes. We answer the question by showing that if the attribute grammars are what we call determinate, then there exists a simple language, generated by a logn-bounded attribute grammar with inherited attributes, which requires more than space n/log n in any attribute grammars with synthesized attributes only. The logn-bounded grammar is L-attributed ([LRS]) and is the "natural" attribute grammar for the language.

The determinacy-requirement ensures that if a string has several derivation trees, then the attribute values in all the trees "make sense". This means that determinate grammars can't be used in a "guess-and-check" fashion, and this restriction is vital for our argument. On the other hand, the class of determinate attribute grammars is sufficiently large to include all well-defined grammars (in the sense of [Knu]) whose underlying context-free grammar is unambiguous.

2. Space-restricted attribute grammars

We follow [EF] in the definition of attribute grammars.

A semantic domain is a pair (Ω, Φ) where Ω is a set of sets (the sets of attribute values) and Φ is a collection of mappings (the semantic functions) of the form $f: V_1 \times V_2 \times \ldots \times V_m \rightarrow V_0$ where $m \geq 0$ and $V_i \in \Omega$ for $0 \leq i \leq m$.

An attribute grammar A over semantic domain (Ω, Φ) consists of 1)-4) as follows:

1) A reduced context-free grammar $G = (N, \Sigma, P, S)$ called the underlying context-free grammar of A.

2) Each nonterminal F in G has two associated finite sets, Sy(F) and In(F), called the synthesized and the inherited attributes of F, respectively. The startsymbol S has no inherited attributes, and one of its synthesized attributes is designated to hold the value or translation of the tree under consideration.

3) With each attribute a is associated a set in Ω, which contains a's values.

4) With each production in G of the form

$$p: F \rightarrow v_0 D_1 v_1 \ldots D_m v_m$$

where $F, D_1, \ldots, D_m$ are nonterminals, is associated a set of semantic rules which define the values of p's applied attributes $Apl(p) = Sy(F) \cup \bigcup_{j=1}^{m} In(D_j)$,

in terms of the values of p's <u>defining</u> attributes $Def(p) = In(F) \cup \bigcup_{j=1}^{m} Sy(D_j)$.
A semantic rule is of the form

$$a \leftarrow f(a_1, \ldots, a_m)$$

where $a \in Apl(p)$, f is a semantic function and the a_i's are either domain values or attributes from Def(p). There is exactly one semantic rule for each applied attribute in every production.

In the following, whenever we refer to G, we always mean the underlying context-free grammar of the attribute grammar under consideration.

In what follows we are interested in viewing attribute grammars as language generators, and of all the different ways in which we can define the language generated by such a grammar (see [Ri]), we take the approach where the semantic rules associated with the nodes in a derivation tree, t, are viewed as a set of equations, E_t, in which the attributes are the unknowns. If we assume that the designated synthesized attribute of the grammar's startsymbol has only two values (which we can denote by <u>true</u> and <u>false</u>), then the language generated by the grammar can be defined as follows.

<u>Definition 1</u> Let A be an attribute grammar over a semantic domain (Ω, Φ) and assume that the value-set associated with the designated synthesized attribute of the startsymbol, d(S), is (<u>true</u>, <u>false</u>). The language generated by the grammar is

$$L(A) = \{w \mid w \in L(G) \text{ and there exists a derivation tree for } w,\ t_w, \text{ such that } E_{t_w} \text{ has a solution in which } d(S) = \underline{true}\}$$ □

This definition allows for the possibility that a word in the language can have several derivation trees, some of whose equations have solutions where d(S) = <u>true</u>, some where d(S) = <u>false</u> and some where there are no solutions. As mentioned in the introduction, we shall restrict attention to the case, where it is sufficient to analyze just one derivation tree of a string, in order to find out whether the string is generated by the attribute grammar. Formally, we define the class of <u>determinate</u> attribute grammars, dAG, to be the class of grammars where, for each word w generated by the underlying context-free grammar, 1) each E_{t_w} has exactly one solution, and 2) the value of d(S) is the same in the solutions to all E_{t_w}'s.

The difference between dAG's, general AG's and socalled <u>unambiguous</u> AG's is discussed further in [Ri]. We now restrict attention to semantic domains (Ω_s, Φ) where Ω_s is the set of all strings over some finite alphabet.

<u>Definition 2</u> Let $\mathcal{C}$ be a class of attribute grammars and let $S: R_+ \to R_+$ be a function mapping nonnegative reals to nonnegative reals. An attribute grammar A (over some domain (Ω_s, Φ)) belongs to $\mathcal{C}(S(n))$ if A is in $\mathcal{C}$, and there exists a constant c such that for every word $w \in L(G)$ and every derivation tree for w, t_w, all attribute values, v(a), in all solutions to E_{t_w} satisfy $|v(a)| \leq c\, S(|w|)$.

□

In the rest of the paper we are only interested in the following two simple classes of attribute grammars:

S-AG: the class of grammars whose nonterminals have no inherited attributes.

L-AG: the class of grammars where an attribute associated with a nonterminal on the righthandside of a production does not depend on any attribute to its right (see [LRS]).

3. Space-restricted L-dAG's and S-dAG's

As described above, we now consider semantic domains (Ω_S, Φ) where Ω_S consists of strings over some finite alphabet. The following theorem shows that inherited attributes add power to space-restricted determinate attribute grammars.

Theorem 3 There exists a language L_0 such that

a) L_0 is generated by an attribute grammar in L-dAG(logn)

b) L_0 is not generated by any attribute grammar in S-dAG(n/logn) □

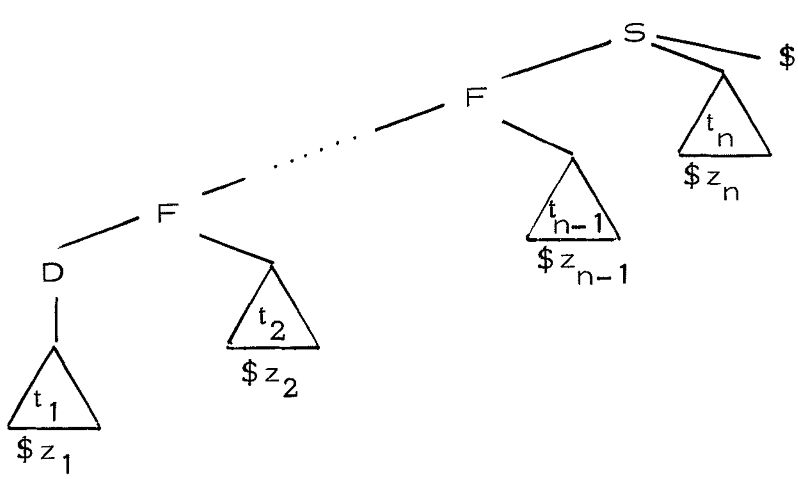
L_0 consists of strings of the form $\$z_1\$\ldots\$z_n\$$ where $z_i \in \{0, 1\}^*$. Each such string is interpreted as a sequence of binary numbers where the substring z_i represents the number $\bar{z}_i$ whose binary representation is $1z_i$. L_0 is defined as follows

$$L_0 = \{\$z_1\$\ldots\$z_n\$ \mid n \geq 2,\ n \text{ even},\ z_i \in \{0,1\}^*,\ \#\{i \mid \bar{z}_i > \tfrac{n}{2}\} \geq \tfrac{n}{2}\}$$

i.e. the language consists of sequences of integers, at least half of which are greater than half the length of the sequence (# M denotes the number of elements in the set M).

Proof of Theorem 3a)

Let A be an attribute grammar (over a domain to be specified later) whose underlying context-free grammar generates the language $(\$\{0,1\}^*\$\{0,1\}^*)^+\$$ in such a way that the derivation tree for the word $z = \$z_1\$\ldots\$z_n\$$ looks as follows



We can specify the semantic rules implicitly by the following top-down left-to-right pass over the tree, which evaluates the attributes.

1. Count the number of trees $t_n, t_{n-1}, \ldots, t_1$ using inherited attributes for F and D.
2. Check whether $\bar{z}_i$ is greater than $\frac{n}{2}$. Each time such a z_i is found, increase a counter (which is a synthesized attribute of D, F and S).
3. Set d(S) = <u>true</u> if the value of the counter is at least $\frac{n}{2}$, otherwise set d(S) = <u>false</u>.

It should be clear that we have specified an L-attributed grammar over some domain (Ω_s, Φ) where Φ contains functions which can compare two values, increase a value by 1 etc. It is also clear that the largest value of the counters is n, and since in step 2 only the $\lceil \log n \rceil$ most significant bits of z_i are needed in the comparison, no attribute in the tree requires more space than $\lceil \log n \rceil$. Since the length of the word z is at least n and since the grammar is obviously determinate, we have shown that L_0 is generated by an attribute grammar in L-dAG(logn). □

Notice that the reason attribute values could be kept small in the above grammar is that the total number of z_i's is known when "processing" each z_i locally. The proof of the second half of the theorem amounts to showing that, when inherited attributes are ruled out and when "guessing" is impossible (this is what determinacy prevents), then the nonterminals in the grammar have to "remember" which substring they generate. The proof is rather lengthy and we present it in the form of a sequence of lemmas.

First, we introduce the following notation. Let z be an arbitrary string of the form $z = \$z_1\$z_2\ldots\$z_n\$$ (where $n \geq 2$). We shall refer to such strings as <u>sequences</u>, to the z_i's as <u>words</u> in z and to the $\bar{z}_i$'s as the <u>numbers</u> in z. Since we want to abstract away from the order in which the numbers in z occur, we define the <u>characteristic vector of z</u> to be the vector

$$\vec{v}(z) = (v_1(z), v_2(z), \ldots, v_{\bar{m}}(z))$$

where $\bar{m} = \max_{1 \leq j \leq n} \{\bar{z}_j\}$ and $v_i(z) = \#\{j \mid \bar{z}_j = i\}$. Hence $v_i(z)$ is the number of j's such that $\bar{z}_j = i$. Using $\vec{v}(z)$, we can characterize the elements of our language L_0 as the set of sequences z, for which

$$\sum_{i=\frac{n}{2}+1}^{\bar{m}} v_i(z) \geq \frac{n}{2}$$

Because of this characterization we are also interested in the <u>accumulated characteristic vector of z</u>, which is denoted by $\vec{s}(z) = (s_0(z), \ldots, s_{\bar{m}-1}(z))$ and defined by

$$s_i(z) = \sum_{j=i+1}^{\bar{m}} v_j(z) \quad \text{for } 0 \leq i \leq \bar{m}-1$$

Now, let, for some $S(n)$, A be an attribute grammar in S-dAG(S(n)) which generates L_0. Because spacebounds are defined up to a constant factor, we can assume without loss of generality that A's underlying context-free grammar G only has productions of the form $F \to DE$, $F \to d$, $F \to \lambda$ where F, D, E are nonterminals, d is a terminal, and λ is the empty word. Since our proof is basically a pumping argument, we need the following "typical" constants. Let m be the number of nonterminals in G, r the number of productions, $k_G = 2^{9m+1}$ and let $k = [(9m+1)k_G]!$ be fixed for the rest of the paper.

We now introduce a structured subset of L_0 for which we can show that the attributes in the grammar must be used to distinguish many words with different characteristic vectors. Let, for each n which is a multiple of k and for which $\lceil \log 3 + \log n \rceil = \lceil \log 4 + \log n \rceil$, K_n be the set

$$K_n = \{\$z_1\$\ldots\$z_{2n}\$ \mid 3n < \bar{z}_i \leq 4n \text{ for } 1 \leq i \leq 2n\}$$

i.e. K_n consists of sequences of length 2n in which each number lies between 3n and 4n. It is clear that K_n is a subset of L_0 and it follows from the construction that all words in sequences in K_n are of length $\lceil \log n \rceil + 1$. We shall call sequences $z_1, z_2 \in K_n$ <u>separable</u> if there exists an $i \geq 0$ such that

$$|s_{ik}(z_1) - s_{ik}(z_2)| \geq 2 \qquad (1)$$

i.e. if there is a component (which is a multiple of k) in their accumulated characteristic vectors where they differ by more than 1. The following lemma shows that there are many separable vectors in K_n (the proof is outlined in the appendix).

<u>Lemma 4</u> There exists an integer n_1 and a constant $c_1 > 0$ such that for all $n \geq n_1$, the number of mutually separable sequences in K_n is at least $2^{c_1 n}$. □

The reason for calling sequences z_1 and z_2 in (1) separable should become clear when we now show that there exists a derivation in the grammar G of the form $S \overset{*}{\Rightarrow} \alpha = sFt$ (s and t are possibly empty terminal strings and F is a nonterminal) such that 1) sFt generates many mutually separable sequences, and 2) if $z_1 = sw_{z_1}t$ and $z_2 = sw_{z_2}t$ are separable then we can pump within s and t to obtain two words $s'w_{z_1}t'$ and $s'w_{z_2}t'$, exactly one of which belongs to L_0.

We construct the form $\alpha = sFt$ iteratively as the limit of a sequence $\alpha_1, \alpha_2, \ldots, \alpha_i, \ldots$ in the following way.

1. $\alpha_1 = S$, where S is the startsymbol of the grammar.

2. Assume that $\alpha_1, \alpha_2, \ldots, \alpha_i$ has been constructed and let $\alpha_i = s_iF_it_i$. Let z be a sequence in K_n generated by α_i, and assume the derivation looks like $\alpha_i = s_iF_it_i \overset{p}{\Rightarrow} s_iDEt_i \overset{*}{\Rightarrow} s_iw_Dw_Et_i = s_iw_zt_i = z$, where $p: F_i \to DE$ is the p'th production in G ($1 \leq p \leq r$). Let us for any such z define its <u>cutpoint</u> to be the length of w_D, and denote by M(p, c) the set of mutually separable sequences in K_n which have cutpoint c and which are generated

by α in the above way (i.e. p is the first production used). Remove from the total collection of sets $M = \{M(p,c) \mid p$ is an F_i-production, $0 \le c \le |w_z|\}$ all sets for which $c = 0$ ($c = |w_z|$) and the form s_iEt_i (s_iDt_i) already occurs in the sequence $\alpha_1, \alpha_2, \ldots, \alpha_i$. Let the remaining collection of sets be M' and choose among these a set M(p, c) which con-contains as many elements as any other set in M'. Let $p(i)$: $F_i \to DE$ and c(i) be the production and cutpoint associated with the chosen set.

3. There are three cases to consider

a) $\underline{c(i) \le (k_G+1)(1+\lceil \log n \rceil)}$. The sequences in M(p(i), c(i)) are of the form $s_iw_Dw_Et_i$ where $|w_D| = c(i)$. Choose w_D such that $s_iw_DEt_i$ generates as many sequences in M(p(i), c(i)) as for any other choice of w_D and let $\alpha_{i+1} = s_iw_DEt_i$. If $s_iw_Dt_i$ contains more than $(9m+1)k_G$ \$'s then <u>halt</u>, otherwise go back to step 2.

b) <u>$c(i) \ge |w_z| - (k_G+1)(1+\lceil \log n \rceil)$. The situation is analogous to a) with C and D interchanged.</u>

c) <u>$(k_G+1)(\lceil \log n \rceil+1) < c(i) < |w_z| - (k_G+1)(1+\lceil \log n \rceil)$.</u> Again the sequences in M(p(i), c(i)) are of the form $s_iw_Dw_Et_i$. Choose among all w_D's and w_E's the one for which $s_iw_DEt_i$ or $s_iDw_Et_i$ generates most sequences in M(p(i), c(i)). Assuming the one chosen is a w_D, let $\alpha_{i+1} = s_iw_DEt_i$ and <u>halt</u>.

Let us say that the above procedure terminates <u>normally</u> if one of the <u>halt</u>-instructions in step 3 is executed.

<u>Lemma 5</u> There exists an integer n_2 and a constant $c_2 > 0$ such that the procedure terminates normally for all $n \ge n_2$ and furthermore the resulting $\alpha = sFt$ generates at least $2^{c_2 n}$ mutually separable sequences in K_n. □

<u>Proof</u>

Let f(i) be the number of mutually separable sequences in K_n generated by α_i. It is clear that any such sequence $z = s_iw_zt_i$ belongs to at least one of the sets in the collection M in step 2 (provided $|w_z| \ge 2$). It is also easy to see that it belongs to at least one of the sets in M', because removing elements with cutpoint 0 (or $|w_z|$) whose "corresponding" sentential form already appeared in the sequence $\alpha_1, \alpha_2, \ldots, \alpha_i$ just amounts to eliminating useless derivations of the form $sFt \stackrel{*}{\Rightarrow} sFt$.

Since no cutpoint can be larger than $2n(\lceil \log n \rceil+3)$ and since the grammar has at most r productions, the set M(p(i), c(i)) above contains at least $\dfrac{f(i)}{2rn(\lceil \log n \rceil+3)}$ elements. If step 3a) or 3b) is executed, the string w_D (w_E) is of length at most $(k_G+1)(1+\lceil \log n \rceil)$ and since there are only 3 terminal symbols in the language, it follows that

$$f(i+1) \geq \frac{f(i)}{2rn(\lceil \log n \rceil + 3) \circ 3^c}$$

for some $c \leq (k_G+1)(1+\lceil \log n \rceil)$. If step 3c) is executed we "loose" at most the square-root of the sequences in $M(p(i), c(i))$, hence

$$f(i+1) \geq \left(\frac{f(i)}{2rn(\lceil \log n \rceil + 3)}\right)^{1/2}$$

Since step 3a) or 3b) is executed at most $i_{max} = m[(9m+1)k_G([\log n]+1)$ times, and step 3c) at most once, we conclude that if the procedure terminates normally then the resulting α generates at least

$$\left(\frac{f(1)}{[(2rn(\lceil \log n \rceil + 3)\ 3^c]^{i_{max}}}\right)^{1/2} \qquad (2)$$

mutually separable sequences in K_n. The dominating term in the denominator is of the form $(n\log n \cdot 3^{\log n})^{c'\log n}$ for some constant c'. But since we know from Lemma 4 that $f(1)$ is (asymptotically) of the form $2^{c_1 n}$ then it follows that (2) is also asymptotically of the form $2^{c_2 n}$ for some $c_2 \geq 0$. Since this implies that each α_i generates lots of separable sequences, we have in fact also shown that the procedure terminates normally.

□

Next we show that if $z_1 = sw_{z_1}t$ and $z_2 = sw_{z_2}t$ are separable sequences from K_n generated by sFt then we can "pump within s and/or t" in such a way that exactly one of the resulting words $s'w_{z_1}t'$ and $s'w_{z_2}t'$ belongs to L_0.

Assume that the iterative procedure above stopped in step 3a) or 3b). We know that st contains at least $(9m+1)k_G$ \$'s and since no step in the construction introduced more than k_G \$'s (otherwise we would have stopped in step 3c)) the path $S = F_1, F_2, \ldots, F_i = F$ from the startsymbol to F contains at least 9m+1 nonterminals each of which generates \$'s. Among these nonterminals there are at least 10 occurrences of the same nonterminal. Choose the largest j such that $F_j, F_{j+1}, \ldots, F_i = F$ contains 10 occurrences of the same nonterminal (generating \$'s). This piece of the path is of the form $B, \ldots, B, \ldots, F$ where B is the "repeating" nonterminal. Hence we have a derivation of the form

$$S \overset{*}{\Rightarrow} uBy \overset{*}{\Rightarrow} uvBxy \overset{*}{\Rightarrow} uvw'Fw''xy = sFt$$

where vx contains at least 9 and at most $(9m+1)k_G$ \$'s.

In the case where the iterative procedure stopped in step 3c), the string w_D added in the last step is itself sufficiently long to allow pumping within the subtree generated by the derivation $D \overset{*}{\Rightarrow} w_D$. The argument is similar to the one just given, which is in fact a trivial extension of the proof of Ogden's lemma found in [AU]. In the last situation we have a derivation of the form

$$S \overset{*}{\Rightarrow} uByFt \overset{*}{\Rightarrow} uvBxyFt \overset{*}{\Rightarrow} uvwxyFt = sFt \qquad (3)$$

where we now know that vx contains at least 9 and at most k_G \$'s.

In the following we shall assume that the derivation looks like (3), the argument in the other case being similar. Let α = uvwxyFt be the sentential form generated in (3) and consider for arbitrary $i \geq 0$ the form

$$\alpha(i) = uv^{i+2}w\, x^{i+2}yFt$$

which is also generated by the grammar. We know that vx contains at least 9 \$'s and we shall assume that both v and x contain at least one \$, the argument in the case where only one of them contains a \$ is similar. If we write v and x in the form $v = a\eta\,\$d$, $x = b\delta\,\$e$ where $\eta, \delta \in (\$\{0,1\}^*)^*$ and $a, b, d, e \in \{0,1\}^*$ then $\alpha(i)$ looks like

$$ua\eta\,\$d\,(a\eta\,\$d)^i a\eta\,\$d\,w\;\; b\delta\,\$e\,(b\delta\,\$e)^i\, b\delta\,\$e\,y\,Ft$$

which we can rewrite as

$$ua\,[\eta\,\$d\,(a\eta\,\$d)^i a]\,\eta\,\$d\,w\;\; b[\delta\,\$e\,(b\delta\,\$e)^i b]\,\delta\,\$e\,y\,Ft \qquad (4)$$

This string is equal to α with the brackets "inserted". We shall show that by choosing i appropriately and by pumping/contracting <u>within</u> the individual pieces $a\eta\,\$d \ldots b\delta\,\$e$,we can distinguish any two separable sequences from K_n generated by α. Let z_1 and z_2 be two such sequences and let $i_0 = ik$ be the index for which $|s_{i_0}(z_1) - s_{i_0}(z_2)| \geq 2$. Assume wlg that

$$s_{i_0}(z_1) > s_{i_0}(z_2) + 1 \qquad (5)$$

and consider the equations

$$\begin{aligned} s_{i_0}(z_1) + p &= i_0 \\ 2n + p + q &= 2i_0 \end{aligned} \qquad (6)$$

Let us call a number which is greater than i_0 a <u>big</u> number and a number which is smaller than or equal to i_0 a <u>small</u> number. If the derivations of z_1 and z_2 are as follows

$$\begin{aligned} S &\overset{*}{\Rightarrow} uvwxyFt \overset{*}{\Rightarrow} uvwxyw_1t = z_1 \\ S &\overset{*}{\Rightarrow} uvwxyFt \overset{*}{\Rightarrow} uvwxyw_2t = z_2 \end{aligned} \qquad (7)$$

and if we can choose i such that the two pieces

$$[\eta\,\$d\,(a\eta\,\$d)^i\, a] \quad \text{and} \quad [\delta\,\$e\,(b\delta\,\$e)^i\, b]$$

together contain p+q words, then we only have to show that we can pump/contract within each of the pieces $a\eta\,\$d \ldots b\delta\,\$e$ in such a way that (almost) p of the words represent big numbers. Because then it follows from (6) that the resulting word $z_1' = uvv_1 \ldots v_i vwxx_i \ldots x_1xyw_1t$ belongs L_0 whereas $z_2' = uvv_1 \ldots v_i vwxx_i \ldots x_1xyw_2t$

does not. Notice that the second equation in (6) says that z_1' contains $2n+p+q = 2i_0$ numbers and that the first equation says that i_0 of these numbers are big. Taken together this means that $z_1' \in L_0$. z_2', on the other hand, also contains $2n+p+q = 2i_0$ numbers but, because of (5), less than i_0 of them are large, i.e. $z_2' \notin L_0$.

Now, the reason the difference in (5) has to be more than 1 is that we can't quite obtain p big and q small words, but as the following lemma says we can do almost as well (the proof is outlined in the appendix).

Lemma 6 There exists an integer n_3 and a constant $c_3 > 0$ such that for all $n \geq n_3$, if $z_1 \in K_n$ is derived as in (7) and if p, q satisfy (6) then there exists a derivation of a word z_1' of the form $z_1' = uvv_1 \ldots v_i vwxx_i \ldots x_1 xyw_1 t$ which contains $2n+p'+q'$ numbers, $s_{i_0}(z_1)+p'$ of which are big, and such that $p'+q' = p+q$ and $p+1 \geq p' \geq p$.

□

Now we can finally prove Theorem 3b).

Proof of Theorem 3b)

Assume that the language L_0 is generated by an attribute grammar in S-dAG(n/logn). Let c_0 be the constant such that for all words w generated by the underlying grammar, any attribute value v(a) in a solution to the equations E_{t_w} satisfies $|v(a)| \leq c_0 (|w|/\log(|w|))$. Let $l(n) = 2n(\lceil \log n \rceil + 2)+1$ and choose n larger than any of the integers n_1, n_2, n_3 in lemmas 4, 5, 6 and such that $c_2 n > c_0 \log(g)(l(n)/\log(l(n)))$ where g is the size of the grammar's attribute-alphabet (recall that attribute values are strings over a finite alphabet). Let α = sFt be the sentential form from Lemma 5 which generates at least $2^{c_2 n}$ mutually separable sequences from K_n. All these sequences are of length l(n), hence the attribute values v(a(F)) associated with the nonterminal F in all these derivations satisfy

$$|v(a(F))| \leq c_0\, l(n)/\log(l(n)) < \frac{c_2}{\log(g)} n$$

The total number of different values a string of length $|v(a(F))|$ can represent is

$$g^{|v(a(F))|} < g^{\frac{c_2 n}{\log g}} = 2^{c_2 n}$$

Hence there are at least two separable sequences in K_n, z_1 and z_2 such that the values of a(F) in the sets of equations $E_{t_{z_1}}$ and $E_{t_{z_2}}$ are equal. Now construct z_1' (and z_2') according to Lemma 6 and consider the corresponding equations $E_{t_{z_1'}}$ and $E_{t_{z_2'}}$. Since the grammar is S-attributed, the set of equations determining a(F) in $E_{t_{z_1'}}$ ($E_{t_{z_2'}}$) is the <u>same</u> as the set of equations determining a(F) in $E_{t_{z_1}}$ ($E_{t_{z_2}}$) because the subtrees with F as root are identical in the two cases. Hence the values of a(F) in $E_{t_{z_1'}}$ and $E_{t_{z_2'}}$ are equal. Since the remaining equations in $E_{t_{z_1'}}$ and $E_{t_{z_2'}}$

are identical, the values of the designated attribute of the startsymbol in the two sets of equations are equal. But that's impossible because z_1' is in the language whereas z_2' is not, and since the grammar is determinate the two values of $d(S)$ must be different. Thus we have reached a contradiction, which shows that no grammar in S-dAG(n/logn) can generate L_0. □

4. Conclusion

Theorem 3 is obtained under very weak assumptions about the semantic domains involved. Indeed in the proof of part 3b) nothing is assumed about the semantic functions. It is also relevant to consider the situation where we, in addition to the restriction on the size of the attribute values, require that the semantic functions belong to some complexity class. The first steps in this direction has been taken in [Je], where semantic functions were measured in terms of the complexity of the RAM-programs used to implement them.

5. Appendix

Here we outline the proofs of Lemma 4 and Lemma 6.

Proof of Lemma 4

Each sequence z in K_n consists of 2n numbers in the range from 3n to 4n. Hence $\vec{v}(z)$ has at most n nonzero components, and the total number of vectors of this form can be computed as the number of ways to distribute 2n balls over n boxes, which is equal to

$$\binom{3n-1}{n-1} \tag{8}$$

Let's say that the accumulated characteristic vectors associated with $z_1, z_2 \in K_n$ are similar if $s_{ik}(z_1) = s_{ik}(z_2)$ for $i \geq 0$. Given an arbitrary sequence $z \in K_n$, the number of vectors similar to $\vec{s}(z)$ is

$$\prod_{i=1}^{\frac{n}{k}} \binom{p_i+k-1}{k-1} \tag{9}$$

where $p_i = s_{(i-1)k}(z) - s_{ik}(z)$, (i.e. $\sum_{i=1}^{\frac{n}{k}} p_i = 2n$).

(9) is maximal when $p_1 = p_2 = \ldots = p_{\frac{n}{k}} = 2k$, which means that the maximal number of vectors similar to any given vector is

$$\binom{3k-1}{k-1}^{\frac{n}{k}} \tag{10}$$

Let's say that $\vec{s}(z_1)$ and $s(z_2)$ are <u>almost similar</u> if $|s_{ik}(z_1) - s_{ik}(z_2)| \leq 1$ for $i \geq 0$. We can bound the number of vectors which are almost similar to any given vector by

$$D_{\frac{n}{k}} \cdot \binom{3k-1}{k-1}^{\frac{n}{k}} \tag{11}$$

where $D_{\frac{n}{k}}$ is a solution to the following difference equations

$$\begin{aligned} D_n &= D_{n-1} + 2\, C_{n-1} \\ C_n &= D_{n-1} + C_{n-1} \\ D_1 &= C_1 = 1 \end{aligned} \tag{12}$$

which are obtained by systematically analyzing the ways in which almost similar vectors can be different at components which are multiples of k.

Dividing (8) by (11) gives a lower bound on the number of separate sequences in K_n. Since the solution to (12) is $D_n \approx (1+\sqrt{2})^n$ and since we can show (using Stirling's Formula) that (8) divided by (10) is asymptotically equal to $3^{\frac{n}{k}}$, we find that for sufficiently large n, the number of mutually separable sequences in K_n is at least $\left(\frac{3}{1+\sqrt{2}}\right)^{\frac{n}{k}}$ which is equal to $2^{c_1 n}$ for some $c_1 > 0$. □

<u>Proof of Lemma 6</u>

Assume that z_1, z_2, p and q satisfy (5) and (6). Since the sequences z_1 and z_2 only contain numbers between 3n and 4n, any index i_0 for which their s-vectors differ, satisfies $3n \leq i_0 < 4n$. Furthermore, the sequences both contain 2n numbers, which means that $0 \leq s_{i_0}(z_1) \leq 2n$. From this it follows that $n \leq p < 4n$, $4n \leq p+q < 6n$ and $\frac{q}{p} \geq \frac{1}{3}$. Finally, since both n and i_0 are multiples of k, so is p+q. Now, consider the piece

$$\ldots v \ldots x \ldots = \ldots a\eta \,\$ d \ldots b\delta \,\$ e \ldots \tag{13}$$

which can be pumped according to (3) and (7). Assume that (13) contains h occurrences of a \$. We know that $9 \leq h \leq k_G$ and since $k = [(9m+1)k_G]!$, h is a divisor in k. Hence there exists a j such that $h \cdot j = (p+q)$, and it follows that the sequence

$$z_1' = uvv^{j-1}vwxx^{j-1}xyw_1t \tag{14}$$

contains $2n+p+q = 2i_0$ words. Now we only have to show that we can make sure that exactly p' of the p+q added numbers are big, where $p+1 \geq p' \geq p$. This is shown by observing that when n is large, then the distance between consecutive \$'s in (13) is large, and since the number of \$'s in (13) is independent of n, we can pump within the piece (13) <u>without</u> changing the number of occurrences of \$'s.

To be precise, we can show the following observation, again by extending the proof of Ogden's Lemma (see [AU]).

Observation Let $G = (N, \Sigma, P, S)$ be a context-free grammar with m nonterminals whose productions are of the form $F \to DE$, $F \to d$, $F \to \lambda$, and let $\Sigma = \Sigma_1 \cup \Sigma_2$ be a partition of Σ (i.e. $\Sigma_1 \cap \Sigma_2 = \emptyset$). If G generates a word of the form $z = u'w'y'$ in which the total number of occurrences of symbols from Σ_2 is g, $w' \in \Sigma_1^*$, and $|w'| \geq 2^{2mg+3}$ then z can be written in the form $z = uvwxy$, where $vx \in \Sigma_1^*$, v (or x) is a nonempty substring of w' and each word of the form uv^iwx^iy ($i \geq 0$) is also generated by G.

□

Using this observation, we can construct three new pumping pieces of the form

$$\ldots a'\eta'\$d' \ldots b'\delta'\$e'$$

one in which all words, except possibly a', d', b', e', are long (and thus represent big numbers), one in which they are all short (and thus represent small numbers), and one in which all but one or two are short. It is easy to show that we can replace $\ldots v^{j-1} \ldots x^{j-1} \ldots$ in (14) by properly chosen copies of these three pieces in such a way that p' words are long and the rest are short, and this proves the lemma. Notice that when using the pumping piece with long words, we might obtain 2 short words every time we get h-2 long ones. But since $\frac{q}{p} \geq \frac{1}{3}$ and $h \geq 9$, that many short words are needed anyway (this is the reason for the occurrence of the number 9 in our various constants).

□

Acknowledgments

The initial inspiration for this work came up during discussions with Poul Jespersen. Ole Lehrmann Madsen's expertise on attribute grammars has also been very helpful.

References

[AU] A.V. Aho and J.D. Ullman [1972]. "The Theory of Parsing, Translation and Compiling" Vol. 1: Parsing, Prentice-Hall, Englewood Cliffs, N.J.

[Dus] J. Duske, R. Parchmann, M. Sedello and J. Specht [1977]. "IO-macrolanguages and attributed translations", Information and Control 35, 87-105.

[EF] J. Engelfriet and G. Filé [1979]. "The formal power of one-visit attribute grammars", Manuscript, Tech. Hoogeschool, Twente, Netherlands.

[Je] P. Jespersen [1979]. "Attributgrammatikker med begrænsede semantiske funktioner", Master's Thesis, Aarhus University, Denmark.

[Knu] D.E. Knuth [1968]. "Semantics of context-free languages", Math. Syst. Theory 2, 127-145.

[LRS] O.M. Lewis, P.J. Rosenkrantz and R.E. Stesrns [1974]. "Attributed translations", JCSS 9, 191-194.

[Ri] H. Riis [1980]. "Subclasses of attribute grammars", Master's Thesis, Aarhus University, Denmark.