ELSEVIER

Theoretical Computer Science 205 (1998) 195–205

Theoretical Computer Science

Characterizations of recursively enumerable languages by means of insertion grammars[1]

Carlos Martin-Vide[a], Gheorghe Păun[b], Arto Salomaa[c,*]

[a] *Universitat Rovira i Virgili, Grup de Recerca en Linguistica Matematica, Pl. Imperial Tarraco, 1, 43005 Tarragona, Spain*

[b] *Institute of Mathematics of the Romanian Academy, P.O. Box 1 – 764, 70700 Bucureşti, Romania*

[c] *Academy of Finland and Turku Centre for Computer Science TUCS Data City, 4th floor, Lemminkaisenkatu 14 A, 20520 Turku, Finland*

Received September 1996
Communicated by M. Nivat

Abstract

An insertion grammar is based on pure rules of the form $uv \to uxv$ (the string x is inserted in the context (u, v)). A strict subfamily of the context-sensitive family is obtained, incomparable with the family of linear languages. We prove here that each recursively enumerable language can be written as the weak coding of the image by an inverse morphism of a language generated by an insertion grammar (with the maximal length of strings u, v as above equal to seven). This result is rather surprising in view of some closure properties established earlier in the literature. Some consequences of this result are also stated. When also erasing rules of the form $uxv \to uv$ are present (the string x is erased from the context (u, v)), then a much easier representation of recursively enumerable languages is obtained, as the intersection with V^* of a language generated by an insertion grammar with erased strings (having the maximal length of strings u, v as above equal to two).

1. Insertion grammars

Most of the generative mechanisms investigated in formal language theory (Thue systems, Post systems, Chomsky grammars, pure grammars, Lindenmayer systems, etc.) are based on the operation of *rewriting*; see, e.g., [13, 14]. However, there are several classes of grammars whose basic ingredient is the *adjoining* operation. The most important of them are the tree adjoining grammars (TAG) [5], the contextual grammars

* Corresponding author. E-mail: asalomaa@kontu.utu.fi.

[1] Research supported by the Spanish Secretaria de Estado de Universidades e Investigacion, SAB95-0357, and the Academy of Finland, Project 11281.

0304-3975/98/$19.00
PII S0304-3975(97)00079-0

[7], and the insertion grammars [4], all three introduced as models of constructions in natural languages.

The insertion grammars (in [4] they are called semi-contextual grammars) are somewhat intermediate between Chomsky context-sensitive grammars (where the nonterminals are rewritten according to specified contexts) and Marcus contextual grammars (where contexts are adjoined to specified strings associated with contexts). In insertion grammars strings are adjoined depending on contexts: one gives triples of the form (u,x,v), defining a substitution of uv by uxv (the adjoining of x in the context (u,v)). Thus, insertion grammars can be also seen as pure grammars whose rules are of the form $uv \to uxv$ (that is, length-increasing pure grammars [8] of a particular form).

Formally, an *insertion grammar* is a triple $G=(V,S,P)$, where V is an alphabet, S is a finite language over V, and P is a finite set of triples of the form (u,x,v), with $u,x,v \in V^*$.

(As usual, we denote by V^* the free monoid generated by the alphabet V under the operation of concatenation; the empty string is denoted by λ. We also denote by *FIN*, *REG*, *LIN*, *CF*, *CS*, *RE* the families of finite, regular, linear, context-free, context-sensitive, recursively enumerable languages, respectively. For other elementary notions of formal language theory, we refer to [14, 13].)

With respect to an insertion grammar $G=(V,S,P)$ we define the relation $\Rightarrow$ on V^* by

$$w \Rightarrow z \text{ iff } w=w_1uvw_2,\ z=w_1uxvw_2 \quad \text{for } (u,x,v)\in P, w_1,w_2 \in V^*.$$

Then, the language generated by G is defined by

$$L(G)=\{z \in V^* \mid w \Rightarrow^* z,\ w \in S\}.$$

Clearly, the insertion rules of the form (u,λ,v) are of no use, hence in what follows we shall assume that no such a rule appears in our grammars.

For an insertion grammar $G=(V,S,P)$ we denote

$$weight(G)=\max\{|u| \mid (u,x,v)\in P, \text{ or } (v,x,u)\in P\}.$$

The family of languages $L(G)$ generated by insertion grammars of weight at most n, $n \geqslant 0$, is denoted by INS_n; the union of all these families is denoted by INS_∞.

Proofs of the following basic results about families of insertion languages can be found in [4, 9, 10, 15].

1. $FIN \subset INS_0 \subset INS_1 \subset \cdots \subset INS_\infty \subset CS$.
2. *REG* is incomparable with all families INS_n, $n \geqslant 1$, and $REG \subset INS_\infty$.
3. $INS_1 \subset CF$, but *CF* is incomparable with all INS_n, $n \geqslant 2$, and INS_∞; INS_2 contains non-semilinear languages.
4. *LIN* is incomparable with all INS_n, $n \geqslant 0$, and INS_∞.

5. All families INS_n, $n \geqslant 0$, are anti-AFLs (that is, they are closed under none of the following operations: union, concatenation, Kleene $*$, direct and inverse morphisms, intersection with regular languages).
6. Each regular language is the morphic image of a language in INS_1.

In view of these poor closure properties (a feature specific to all rewriting systems not using nonterminal symbols), it is of interest to look for the smallest AFL (or related structure) containing a given family INS_n, $n \geqslant 0$. As we shall see in the following section, the result is unexpected: the closure of INS_7 under direct and inverse morphisms is equal to the family of recursively enumerable languages. Contrast this with the fact that all families INS_n are incomparable with LIN. Taking into account that an insertion grammar just adds symbols to the currently generated string, hence the capability to *change* the string looks quite restricted, our characterization of RE is rather surprising. Our result bears some similarity to the characterizations of RE by contextual languages in [3, 2], but note that in [3] pairs of strings are adjoined, hence we can easily mark substrings u of the current string where type-0 Chomsky rules $u \to v$ are simulated, whereas in [2] one uses infinitely many rules, under the form of context-free selectors associated with contexts. These differences between insertion grammars and the contextual grammars used in [3, 2] make new proof techniques necessary, leading to more complex constructions in the case of insertion grammars.

2. A characterization of RE

Theorem 1. *For each language $L \in RE$ there are a morphism h, a weak coding g, and a language $L' \in INS_7$ such that $L = g(h^{-1}(L'))$.*

Proof. Consider a language $L \subseteq T^*$, $L \in RE$, generated by a type-0 Chomsky grammar $G = (N, T, S, P)$ in Kuroda normal form, that is with the rules in P of the following types:

1. $A \to BC$, $A \to a$, $A \to \lambda$, for $A, B, C \in N, a \in T$,
2. $AB \to CD$, for $A, B, C, D \in N$.

From the form of the rules, we may assume that each string in $L(G)$ is generated by a derivation consisting of two phases, one when only nonterminal rules are used and one when only terminal rules are used; moreover, we may assume that during the second phase the derivation is performed in the leftmost mode.

Consider the new symbols $\#, \$, c$ and construct the insertion grammar

$$G' = (N \cup T \cup \{\#, \$, c\}, \{c^4 S c^6\}, P'),$$

with P' containing the following insertion rules.

(1) For each context-free rule $r : A \to x \in P$ we consider the rules

(1.r) $(\alpha_1\alpha_2\alpha_3\alpha_4 A, \#\$x, \alpha_5\alpha_6\alpha_7\alpha_8\alpha_9\alpha_{10})$, for $\alpha_i \in N \cup \{\#, \$, c\}$, $1 \leqslant i \leqslant 10$, $\alpha_3\alpha_4 \notin N\{\$\}, \alpha_2\alpha_3\alpha_4 \notin N\{\$\}N$, $\alpha_1\alpha_2\alpha_3\alpha_4 \notin N\{\$\}NN$, $\alpha_5 \notin \{\#, \$\}$, and if $\alpha_5\alpha_6\alpha_7\alpha_8\alpha_9 \in N\{\#\$\}N\{\#\}$, then $\alpha_{10} \in N \cup \{c\}$.

(2) For each non-context-free rule $r : AB \to CD \in P$ we consider the rules

(2.r.1) $(\alpha_1\alpha_2\alpha_3 A, \$CD, B\alpha_4)$, for $\alpha_i \in N \cup \{\#, \$, c\}, 1 \leqslant i \leqslant 4$, and $\alpha_1\alpha_2\alpha_3 \notin N\{\$\}N$, $\alpha_2\alpha_3 \notin N\{\$\}$, $\alpha_4 \notin \{\#, \$\}$,

(2.r.2) $(A\$CDB, \#\$, \alpha)$, for $\alpha \in N \cup \{c\}$,

(2.r.3) $(A, \#, \$CDB\#\$)$.

(3) For each $A, B \in N$ we consider the rules

(3.AB.1) $(\alpha_1\alpha_2\alpha_3 AB\#\$, A\#, \alpha_4\alpha_5\alpha_6)$, for $\alpha_i \in N \cup \{\#, \$, c\}$, $1 \leqslant i \leqslant 6$, $\alpha_1\alpha_2\alpha_3 \notin N\{\$\}N$, and if $\alpha_4\alpha_5 = A\#$, then $\alpha_6 = \$$.

(3.AB.2) $(A, \#\$, B\#\$A\#\alpha)$, for $\alpha \in N \cup \{c\}$.

(3.AB.3) $(\$B\#\$A\#, \$A, \alpha)$, for $\alpha \in N \cup \{c\}$.

We say that all rules (1.r) are *of type* 1, all rules (2.$r.i$), for r a non-context-free rule in P and $1 \leqslant i \leqslant 3$, are *of type* 2, and that all rules (3.$AB.i$), $A, B \in N$ and $1 \leqslant i \leqslant 3$, are *of type* 3.

Denote by U the set of strings $\alpha\#\$$, for $\alpha \in N \cup T$. For each string $w \in U$ we consider a symbol b_w. Let W be the set of these symbols. We define the morphism

$$h : (W \cup T \cup \{c\})^* \to (N \cup T \cup \{\#, \$, c\})^*$$

by

$$h(b_w) = w, \quad w \in U,$$
$$h(a) = a, \quad a \in T,$$
$$h(c) = c.$$

Consider also the weak coding

$$g : (W \cup T \cup \{c\})^* \to T^*,$$

defined by

$$g(b_w) = \lambda, \quad w \in U,$$
$$g(c) = \lambda,$$
$$g(a) = a, \quad a \in T.$$

We obtain

$$L(G) = g(h^{-1}(L(G'))).$$

The intuition behind the construction above is the following.

The insertion rules of type 1 simulate the context-free rules of G, the rules of type 2 simulate the non-context-free rules of G. The rules of type 3 are used in order to prepare the current string for making possible the use of rules of type 2. This is done as follows.

The symbols $\#, \$$ are called *markers*. A nonterminal followed by # and then by a symbol different from \$ is said to be *#-marked*. A nonterminal followed by \$ is said to

be \$-*marked*. A nonterminal followed by #\$ is said to be #\$-*marked*. A nonterminal which is #-, \$-, or #\$-marked is said to be *marked*, otherwise it is called *unmarked*. A string consisting of unmarked symbols in $N \cup T \cup \{c\}$ and of blocks $\alpha\#\$$, for $\alpha \in N \cup T$, is said to be *legal*.

For example, c^4Sc^6 (the axiom of G') is legal, $cX\#\$XaY\#\$c$ is also legal. The first occurrence of X and the occurrence of Y in this latter string are marked (#\$-marked), the second occurrence of X, as well as all occurrences of c and a are unmarked. However, $cX\$XaY\#\$c$ is not legal, because the first occurrence of X is \$-marked but not #\$-marked.

Now, the rules of type 3 are able to move an unmarked nonterminal A across a block $X\#\$$ placed immediately to the right of A. In this way, pairs AB can be created, needed for simulating the context-sensitive rules of G.

The marked symbols, plus the markers and the symbol c are considered an "invisible garbage"; at each moment, the string of the unmarked symbols are intended to correspond to a sentential form of G. By the definitions of h and g, this "invisible garbage" is erased, indeed, from each legal string generated by G'. Because no unmarked nonterminal can be mapped by h^{-1}, what remains will be a terminal string.

In order to prove the equality $L(G) = g(h^{-1}(L(G')))$ we shall first prove that rules in groups 1, 2, 3 in G' are doing what we have said that they are supposed to do (in this way we obtain the inclusion $\subseteq$), then we shall prove that they cannot do anything else (that is also $\supseteq$ is true).

Claim 1. *When using a rule* $(\alpha_1\alpha_2\alpha_3\alpha_4 A, \#\$x, \alpha_5\alpha_6\alpha_7\alpha_8\alpha_9\alpha_{10})$ *of type* 1, *the occurrence of* A *in the derived string is unmarked, but it is* #\$*-marked in the resulting string, where also each symbol of* x *is unmarked.*

The fact that A is unmarked in the string to which the rule is applied is ensured by α_5, which is different from # and \$. Because we obtain the substring $A\#\$x\alpha_5$, the other assertions are obvious.

Claim 2. *When using a group of rules* $(2.r.i)$, $1 \leqslant i \leqslant 3$, *associated with a rule* $r : AB \to CD$ *in* P, *then the symbols* AB *are unmarked in the derived string, both of them will be* #\$*-marked in the resulting string, where also* CD *are unmarked.*

The substring of the string to which the rule $(2.r.1)$ is applied is $\alpha_1\alpha_2\alpha_3 AB\alpha_4$, with $\alpha_4 \notin \{\#, \$\}$, hence A and B are unmarked. We get the string $\alpha_1\alpha_2\alpha_3 A\$CDB\alpha_4$, to which the rule $(2.r.2)$ is applied, leading to $\alpha_1\alpha_2\alpha_3 A\$CDB\#\$\alpha_4$. Now, by the third rule, we get $\alpha_1\alpha_2\alpha_3 A\#\$CDB\#\$\alpha_4$. One sees how the third rule completes the #\$-marking of A, whereas B has been #\$-marked by the second rule. Clearly, CD are always unmarked. From a substring where the only unmarked block (not involving the substrings $\alpha_1\alpha_2\alpha_3$ and α_4) is AB we have obtained a substring where the only unmarked block (not involving the substrings $\alpha_1\alpha_2\alpha_3$ and α_4) is CD.

Claim 3. *Starting from a legal string, the rules in a group* $(3.AB.i)$, $1 \leqslant i \leqslant 3$, *can replace a substring* $AB\#\$\alpha$ *(hence with an unmarked* A*) by a substring consisting of blocks in* $N\{\#\$\}$ *and ending with* $A\alpha$ *(hence with an unmarked* A*).*

The first rule, $(3.AB.1)$, can be applied to a string $x\alpha_1\alpha_2\alpha_3 AB\#\$\alpha_4\alpha_5\alpha_6 y$ and it produces the string $x\alpha_1\alpha_2\alpha_3 AB\#\$A\#\alpha_4\alpha_5\alpha_6 y$. The second rule is now applicable, leading to $x\alpha_1\alpha_2\alpha_3 A\#\$B\#\$A\#\alpha_4\alpha_5\alpha_6 y$. Finally, the third rule produces $x\alpha_1\alpha_2\alpha_3 A\#\$B\#\$A\#\$A\alpha_4\alpha_5\alpha_6 y$. Therefore, the substring $AB\#\$$ has been replaced by $A\#\$B\#\$A\#\$A$, having an unmarked A on the rightmost position.

Thus, starting from a legal string (initially, we have c^4Sc^6), the rules of G' can simulate the rules of G, producing legal strings. Moreover, if we denote by $umk(x)$ the string of the unmarked symbols in a legal string x generated by G', then we have

Claim 4. (i) *If* $x \Rightarrow^* y$ *by using a rule in group* 1 *or all three rules* $(2.r.i)$, $1 \leqslant i \leqslant 3$, *associated with a non-context-free rule* r *of* G, *then* $umk(x) \Rightarrow umk(y)$ *by the corresponding rule in* G.

(ii) *If* $x \Rightarrow^* y$ *by using the three rules in group* 3 *associated to the same* A, B *in* N, *then* $umk(x) = umk(y)$.

Claim 5. *If* $x = g(h^{-1}(y))$, *for some* $y \in L(G')$, *then* y *is a legal string and* $x = umk(y), y \in T^*$. *Conversely, if* $y \in L(G')$ *and* $umk(y) \in T^*$, *then* $umk(y) = g(h^{-1}(y))$.

This follows immediately from the definitions of the morphisms g and h.

These claims prove the inclusion $L(G) \subseteq g(h^{-1}(L(G')))$.

We shall now show that only derivations as above lead to legal strings.

Claim 6. *After using a rule* $(2.r.1)$, *no other rule but* $(2.r.2)$ *can be applied to the involved nonterminals* A, B, C, D. *Then, after* $(2.r.2)$, *only* $(2.r.3)$ *can be used.*

Indeed, let us consider only the subword $\alpha_1\alpha_2\alpha_3 AB\alpha_4$ used by a rule $(2.r.1)$, for $r : AB \to CD \in P$. After using $(2.r.1)$ we obtain $\alpha_1\alpha_2\alpha_3 A\$CDB\alpha_4$. Now:

- No rule $(1.q)$ can be used to any of A, B, C, D, due to the symbols β_i, $1 \leqslant i \leqslant 10$, in rules $(\beta_1\beta_2\beta_3\beta_4 X, \#\$x, \beta_5\beta_6\beta_7\beta_8\beta_9\beta_{10})$ of type $(1.q), q : X \to x \in P$. (For instance, $\beta_2\beta_3\beta_4 \notin N\{\$\}N$, hence D above cannot be used by a rule $(1.q)$ corresponding to $q : D \to x \in P$.)
- No rule $(2.q.1)$ can be used for a pair CD or DB, due to symbols $\beta_1\beta_2\beta_3$ in rules $(\beta_1\beta_2\beta_3 X, \$YZ, U\beta_4)$ of type $(2.q.1)$ for $q : XU \to YZ \in P$.
- No rule $(2.q.2), q \neq r$, can be used: this is obvious, because we need the subword $A\$CDB$, which identifies the rule r in P.
- No rule $(2.q.3)$ can be used, because we need a substring $A\$CDB\#\$$, and α_4 above is different from #.

– No rule (3.*XY*.1) can be used, because our string does not contain the substring *XY*#\$; the same argument makes impossible the use of the rules (3.*XY*.2) and (3.*XY*.3), for all $X, Y \in N$.

Using the rule (2.*r*.2) we get the string $\alpha_1\alpha_2\alpha_3 A\$CDB\#\$\alpha_4$. Nothing has been changed to the left of *A*\$*CDB* or inside this substring; moreover, *B* is now #\$-marked. As above, one can see that no rule can be applied to this string, excepting (2.*r*.3). For instance:

– No rule (3.*DB*.1) can be used for the pair *DB* (the only one which is followed by #\$), because $\beta_1\beta_2\beta_3$ in a rule $(\beta_1\beta_2\beta_3 DB\#\$, D\#, \beta_4\beta_5\beta_6)$ of this type cannot be *A*\$*C*.

– No rule (3.*XY*.2) can be used, because there is no symbol *X* which is #-marked in our string; the same reason makes impossible the use of a rule (3.*XY*.3), $X, Y \in N$.

Claim 7. *After using a rule* (3.*AB*.1), *no other rule but* (3.*AB*.2) *can be applied to the involved nonterminals A, B. Then, after using* (3.*AB*.2), *no other rule than* (3.*AB*.3) *can be used.*

The rule (3.*AB*.1) replaces a substring $\alpha_1\alpha_2\alpha_3 AB\#\$\alpha_4\alpha_5\alpha_6$ by $w = \alpha_1\alpha_2\alpha_3 AB\#\$A\#\alpha_4\alpha_5\alpha_6$. Now:

– No rule of type (1.*q*) : $(\beta_1\beta_2\beta_3\beta_4 A, \#\$x, \beta_5\beta_6\beta_7\beta_8\beta_9\beta_{10})$ can be used (*A* is the only unmarked symbol in our string), because of $\beta_5\beta_6\beta_7\beta_8\beta_9\beta_{10}$ which cannot be equal to $B\#\$A\#\alpha_4$.

– No rule of type (2.*q*.1) can be used, because we do not have two unmarked symbols in *w*.

– No rule of types (2.*q*.2), (2.*q*.3) can be used, because we do not have a \$-marked symbol in *w*.

– No rule (3.*XY*.1) : $(\beta_1\beta_2\beta_3 XT\#\$, X\$, \beta_4\beta_5\beta_6)$ can be used; the only possibility is to use again (3.*AB*.1) (no other symbols appear here), but $\beta_4\beta_5\beta_6$ prevents that.

– No rule (3.*XY*.2) with $XY \neq AB$ can be used, just because we do not have the necessary occurrences of *x* and *Y*.

– No rule (3.*XY*.3) can be used, because we need a substring of the form \$*Y*#\$*X*#, and such a substring does not appear in *w*.

Therefore, we have to continue with (3.*AB*.2) and we get the string $\alpha_1\alpha_2\alpha_3 A\#\$B\#\$A\#\alpha_4\alpha_5\alpha_6$. There is no unmarked symbol here, hence rules of the forms (1.*r*), (2.*q*.1), (2.*q*.2), (2.*q*.3), (3.*XY*.1), (3.*XY*.2) cannot be used. A rule (3.*XY*.3) can be used only if $XY = AB$, which concludes the proof of Claim 7.

Consequently, the rules in groups (1.*r*), for *r* a context-free rule of *P*, and (2.*r.i*), $1 \leqslant i \leqslant 3$, for *r* a non-context-free rule of *P*, and (3.*AB.i*), $1 \leqslant i \leqslant 3$, for $A, B \in N$, cannot be mixed; inside these groups, the rules have to be used in the order of *i*, from 1 to 3, therefore, the grammar *G'* can only simulate derivations in *G* on unmarked symbols. This means that if h^{-1} is defined for $y \in L(G')$, then $c^4 S c^6 \Rightarrow^* umk(y)$ in the grammar *G* and $g(h^{-1}(y)) \in L(G)$, proving the inclusion $g(h^{-1}(L(G'))) \subseteq L(G)$.

Note that $weight(G') = 7$. □

3. Some consequences

Let us denote by $CH^{-1}(F)$ the family of languages of the form $g(h^{-1}(L))$, where g is a weak coding, h is a morphism, and $L \in F$, for a given family of languages, F. With this notation, from the previous theorem and the obvious inclusion $CH^{-1}(F) \subseteq RE$ for all $F \subseteq RE$ (the Turing–Church thesis), we can write

Corollary 1. *$RE = CH^{-1}(INS_n)$, for all $n \geqslant 7$.*

The proof of Theorem 1 in Section 2 can be modified as follows:

- Write $L = (L \cap \{\lambda\}) \cup \bigcup_{a \in T} (\partial_a^r(L)\{a\})$, where ∂_x^r is the right derivative with respect to x, and take a grammar $G_a = (N_a, T, S_a, P_a)$ for each language $\partial_a^r(L)$. Assume the alphabets N_a, $a \in T$, mutually disjoint.
- Start from the axiom set $\{d\}(L \cap \{\lambda\}) \cup \{c^4 S_a c a \mid a \in T\}$.
- Together with all rules in the construction in Section 2 associated with rules in P_a, $a \in T$, consider also the rules with the "witness" suffixes of the type $\alpha_1 \ldots \alpha_k$ ending with the symbol c. For instance, together with

  $(1.r)$: $(\alpha_1\alpha_2\alpha_3\alpha_4 A, \#\$x, \alpha_5\alpha_6\alpha_7\alpha_8\alpha_9\alpha_{10})$,

  consider also all rules with $\alpha_5\alpha_6\alpha_7\alpha_8\alpha_9\alpha_{10}$ replaced by

  $\alpha_5\alpha_6\alpha_7\alpha_8\alpha_9 c$, $\alpha_5 \in \{\#, \$\}$,

  $\alpha_5\alpha_6\alpha_7\alpha_8 c$,

  $\alpha_5\alpha_6\alpha_7 c$,

  $\alpha_5\alpha_6 c$,

  $\alpha_5 c$, $\alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_8, \alpha_9 \in N \cup \{\$\}$,

  c.

Similarly for rules of all other types which involve suffixes of symbols α.

In this way, at the end of the current string we can use shortened rules and still we can prevent the derivations which can produce strings outside the languages $\partial_a^r(L)$.

- Allow also to the terminal symbols to migrate to the right, by rules in group 3, hence let A and B in these rules be also terminals; moreover, let B be also equal to c.
- Add the following rules:

  $(4.a.1)$: $(ac, \#\$a\#, b)$, $a, b \in T$,

  $(4.a.2)$: $(a, \#\$, c\#\$a\#, b)$, $a, b \in T$,

  $(4.a.3)$: $(\$c\#\$a\#, \$ca, b)$, $a, b \in T$.

Note the fact that the symbol c existing in the string is now #\$-marked and that together with the unmarked occurrence of a moved to the right we introduce an unmarked

occurrence of c. The derivation steps are

$$xacbx' \Rightarrow xac\#\$a\#bx' \Rightarrow xa\#\$c\#\$a\#bx' \Rightarrow xa\#\$c\#\$a\#\$cabx',$$

hence the symbol a has been moved to the left of the terminal b.
– Add also the rule

$$(4.a.4)\text{: } (\$c\#\$a\#, \$da, b), \quad a, b \in T,$$

where d is a new symbol, which is introduced in the alphabet of G'.
Because the rule $(4.a.1)$ uses an unmarked occurrence of c, if we use the rule $(4.a.4)$ instead of $(4.a.3)$, then we introduce no new unmarked occurrence of c, hence rules $(4.a.i)$ can no longer be applied. Therefore, if we consider the regular language

$$R = \{\alpha\#\$ \mid \alpha \in (N \cup T \cup \{c\})^*\}\{d\},$$

then we obtain the equality

$$L = R \backslash L(G').$$

Indeed, the left quotient with R selects from $L(G')$ those strings which contain the symbol d and which have in front of this symbol only #\$-marked symbols. This means that all nonterminals were replaced by terminals and that all terminals were moved to the right, hence a copy of them is now present to the right of d. Consequently, we obtain

Corollary 2. *For all $n \geqslant 7$, each language $L \in RE$ can be written in the form $L = R \backslash L'$, for R a regular language and $L' \in INS_n$.*

It is an *open problem* whether or not the "magical number seven" appearing here can be replaced by a smaller one. Anyway, from $INS_1 \subseteq CF$ we have $CH^{-1}(INS_1) \subseteq CF$, because CF is closed under inverse morphisms and arbitrary morphisms. Therefore, the subscript 7 above cannot be replaced by 0 or by 1.

A quite interesting consequence about the size of families INS_n, $n \geqslant 7$, can be inferred:

Corollary 3. *Each family INS_n, $n \geqslant 1$, INS_∞, is incomparable with each family F such that $LIN \subseteq F \subset RE$ and F is closed under weak codings and inverse morphisms, or under left quotients with regular languages.*

Proof. Because $LIN - INS_\infty \neq \emptyset$, we get $F - INS_\infty \neq \emptyset$. Because $CH^{-1}(F) \subseteq F \subset RE$, we cannot have $INS_n \subseteq F$, for $n \geqslant 7$, or $INS_\infty \subseteq F$ (then $RE = CH^{-1}(INS_n) \subseteq F \subset RE$, a contradiction). □

Important families of languages having the properties of F above are MAT^λ (of languages generated by context-free matrix, programmed, regularly controlled, etc. grammars without appearance checking, possibly using λ-rules, see [1]) and $ET0L$

(see [12]). It follows that $INS_n - MAT^{\lambda} \neq \emptyset$, $n \geq 7$. As $INS_{\infty} \subset CS$, we get the fact that $CS - MAT^{\lambda} \neq \emptyset$, a relation which was open for a long time and only recently proved.

4. Insertion-erasing grammars

Following the model of contextual grammars with erased contexts as considered in [11], we can also consider insertion grammars with *erased* strings. Specifically, we can define systems of the form $G=(V,S,P_I,P_E)$, where V is an alphabet, S is a finite set of strings over V, P_I and P_E are finite sets of triples $(u,x,v), u,x,v \in V^*$. The triples in P_I are *insertion rules*, those in P_E are *erasing rules*. They are used in derivations as follows: for $w,z \in V^*$ we write $w \Rightarrow z$ iff one of the two cases below holds:

1. $w=w_1uvw_2, z=w_1uxvw_2$, for $(u,x,v) \in P_I, w_1,w_2 \in V^*$,
2. $w=w_1uxvw_2, z=w_1uvw_2$, for $(u,x,v) \in P_E, w_1,w_2 \in V^*$.

Then, we define the language generated by G as usual,

$$L(G)=\{z \in V^* \mid w \Rightarrow^* z, \text{ for some } w \in S\}.$$

Let us denote by $INSDEL_n$, $n \geq 1$, the family of language as above, generated by insertion grammars with erased strings of weight at most n.

Such mechanisms are very powerful, confirming the general observation that context-sensitivity and erasing can produce everything. Thus, the following result is as expected.

Theorem 2. *Each language $L \in RE, L \subseteq V^*$, can be written in the form $L=L' \cap V^*$, for some $L' \in INSDEL_2$.*

Proof. Take $L \subseteq V^*$ generated by a grammar $G=(N,V,S,P)$ in Kuroda normal form. We construct the insertion grammar

$$G'=(N \cup V \cup \{Y,X_1,X_2\}, \{SYY\}, P_I, P_E),$$

$$P_I=\{(A,X_1x,\alpha_1\alpha_2) \mid A \to x \in P, \alpha_1,\alpha_2 \in N \cup T \cup \{Y\}\}$$
$$\cup\{(AB,X_2CD,\alpha_1\alpha_2) \mid AB \to CD \in P, \alpha_1,\alpha_2 \in N \cup T \cup \{Y\}\},$$

$$P_E=\{(\lambda,AX_1,\lambda) \mid A \in N\} \cup \{(\lambda,ABX_2,\lambda) \mid A \in N\} \cup \{(\lambda,YY,\lambda)\}.$$

The rules in P_I simulate the rules of P. The symbols X_1,X_2 are *markers* of the symbols placed immediately to their left hand: X_1 marks one symbol, X_2 marks two symbols. Due to the right hand member of the contexts of rules in P_I, that is $\alpha_1\alpha_2$, these rules cannot be applied in such a way to involve in their left context a symbol which is already marked. On the other hand, the markers plus the symbols they mark can be erased by the rules of P_E. At the right hand of the string, the correct use of rules in P_I is ensured by the auxiliary symbol Y, whose occurrences can be erased when they are no longer necessary. Consequently, $L=L(G') \cap V^*$. □

The use of erasing rules is very useful: compare the previous proof with the proof of Theorem 1, where, without having erasing possibilities, we needed a procedure for changing the place of unmarked symbols, moving them across the marked ones.

References

[1] J. Dassow, Gh. Păun, Regulated Rewriting in Formal Language Theory, Springer, Berlin, 1989.
[2] A. Ehrenfeucht, A. Mateescu, Gh. Păun, G. Rozenberg, A. Salomaa, On representing RE languages by one-sided internal contextual languages, Tech. Report 95-04, Dept. of Computer Sci., Leiden Univ., 1995; Acta Cybernetica, to appear.
[3] A. Ehrenfeucht, Gh. Păun, G. Rozenberg, On representing recursively enumerable languages by internal contextual languages, Theoret. Comput. Sci., to appear.
[4] B.S. Galiukschov, Semicontextual grammars, Mat. Logica Mat. ling, (1981) 38–50 (in Russian).
[5] A.K. Joshi, L.S. Levy, M. Takahashi, Tree adjunct grammars, J. Comput. System Sci. 19 (1975) 136–163.
[6] L. Kari, G. Thierrin, Contextual insertion/deletion and computability, submitted.
[7] S. Marcus, Contextual grammars, Rev. Roum. Math. Pures Appl. 14 (1969) 1525–1534.
[8] H.A. Maurer, A. Salomaa, D. Wood, Pure grammars, Inform. Control 44 (1980) 47–72.
[9] Gh. Păun, On semicontextual grammars, Bull. Math. Soc. Sci. Math. Roumanie 28 (76) (1984) 63–68.
[10] Gh. Păun, Two theorems about Galiukschov semicontextual grammars, Kybernetika 21 (1985) 360–365.
[11] Gh. Păun, G. Rozenberg, A. Salomaa, Contextual grammars: erasing, determinism, one-sided contexts, in: G. Rozenberg, A. Salomaa, (Eds.), Developments in Language Theory, World Scientific Publ., Singapore, 1994, 370–388.
[12] G. Rozenberg, A. Salomaa, The Mathematical Theory of L Systems, Academic Press, New York, 1980.
[13] G. Rozenberg, A. Salomaa (Eds.), The Handbook of Formal Languages, Springer, Berlin, in preparation.
[14] A. Salomaa, Formal Languages, Academic Press, New York, 1973.
[15] C.C. Squier, Semicontextual grammars: an example, Bull. Math. Soc. Sci. Math. Roumanie 32 (80) (1988) 167–170.