Fundamenta Informaticae 19 (1993) 355–370
IOS Press

DELETION SETS

Lila KARI, Alexandru MATEESCU, Arto SALOMAA
Academy of Finland and University of Turku, Department of Mathematics, 20500 Turku, Finland

Gheorghe PĂUN
Institute of Mathematics of the Romanian Academy of Sciences, Str. Academiei 14, 70109 Bucuresti, Romania

Abstract. We discuss questions related to the cardinality, the effective construction, and decidability of the so-called deletion sets : the sets of strings obtained by erasing from a word the subwords which appear as elements of a given language.

1. Introduction

The operations of inserting or deleting symbols or strings in/from a given string (and the natural extension of such operations to languages) are most fundamental in formal language theory and combinatorics of words. The present paper lies somewhere in between these two fields, as it deals with the deletion operation in the particular case when one starts from a given string w and deletes from it substrings which belong to a given language, L. The result – obviously finite – is denoted by $w \rightarrow L$ and called *deletion set*. The notion is introduced in [2], where one investigates many related operations (sequential, iterated, controlled and scattered insertion and deletion, dipolar deletion etc.) and the following somewhat surprising result is proved: given a set F consisting of two elements only and an arbitrary context-free language L, it is undecidable whether a string w exists such that F can be obtained by deleting from w strings of L (in the previous notation, $F = w \rightarrow L$).

Here we examine more systematically the deletion sets, mainly considering the following four problems: how w and L can be constructed such that $w \rightarrow L$ equals a given deletion set, how large can $w \rightarrow L$ be depending on the structure of w, characterizations of deletion sets, decidability questions of the type of the result in [2] quoted above.

The results show that, although simple at first sight, the deletion sets have intriguing unexpected properties. Let us mention here only the fact that for *any* fixed deletion set F containing at least one non-empty string the following problem is undecidable. Given a context-free language L, determine whether or not a string w exists such that $F = w \rightarrow L$. (Therefore the result in [2] holds for any nontrivial deletion set, including those of the form $\{a\}, a$ being a symbol.)

2. Notations and Terminology

In general, we refer to [4] for basic elements of formal language theory. We specify here only some notations.

For a vocabulary V, we denote by V^* the free monoid generated by V under the operation of concatenation; the empty string is denoted by λ and the length of $x \in V^*$ by $|x|$. For $x \in V^*, a \in V$, we denote by $|x|_a$ the number of occurrences in x of the symbol a. For a finite set F of strings we also denote by $|F|$ the largest length of strings in F, $|F| = max\{|w| \mid w \in F\}$.

For a language $L \subseteq V^*$ we denote by $Pref(L), Suf(L), Sub(L)$ the sets of prefixes, suffixes, subwords, respectively, of strings in L (including λ and the strings themselves). The families of regular and of context-free languages are denoted by REG, CF, respectively.

For $u, v \in V^*$ we denote, following [2],

$$u \rightarrow v = \{z \in V^* \mid u = z_1 v z_2, z = z_1 z_2, z_1, z_2 \in V^*\},$$

and we extend this operation to languages in the natural way:

$$L_1 \rightarrow L_2 = \{z \in (u \rightarrow v) \mid u \in L_1, v \in L_2\}.$$

We are interested here in the particular case when L_1 is a singleton. Namely, we call *deletion sets* the languages F for which a string w and a language L exist such that $F = w \rightarrow L$.

3. Examples and Cardinality Results

Clearly,

$$|w \rightarrow L| \leq |w|,$$

hence each deletion set is finite. In fact, we have

$$(w \rightarrow L) \subseteq Pref(w)Suf(w).$$

On the other hand, not all finite languages are deletion sets. For instance,

$$F = \{a, b, c\},$$

cannot be a deletion set: if $w \rightarrow F = \{a, b, c\}$, then a, b, c must appear as the leftmost and/or the rightmost letters of w and this is impossible. Similarly,

$$F = \{ab, ba, aa, bb\}$$

is not a difference set. In general, we have

Lemma 1. *For every string w and integer $k, 0 \leq k \leq |w| - 1$,*

$$card((w \rightarrow L) \cap V^k) \leq k + 1.$$

Proof. Every string $z \in w \rightarrow L, |z| = k$, is of the form $z = w_1 w_2$, for $w = w_1 v w_2$, $v \in L$, $w_1, w_2 \in V^*$. Clearly, $0 \leq |w_1| \leq k$, and for each choice of w_1, the string w_2 is precisely determined. As we can have $k+1$ possibilities to choose w_1, the relation in lemma is proved (some strings $w_1 w_2, w_1' w_2'$, for $w = w_1 v w_2 = w_1' v' w_2'$, $v, v' \in L$, $|v| = |v'|$, might be equal). ♣

For a given deletion set F, denote

$$M(F) = \frac{(|F|+1)(|F|+2)}{2}.$$

Theorem 1. *For every deletion set F, we have $card(F) \leq M(F)$, and this bound can be reached for every value of $|F|$.*

Proof. Let k range over $0, 1, \ldots, |F|$, in the previous lemma. We obtain $card(F) \leq M(F)$.

Moreover, consider the deletion set

$$F = a^m b^m \rightarrow \{a^i b^j \mid i + j \geq m, 0 \leq i, j \leq m\}.$$

Clearly, $|F| = m$, and for a given value for $i, 0 \leq i \leq m$, we have $i+1$ possible values for j and all of them lead to different strings in F, hence $card(F) = (m+1)\ (m+2)/2$.♣

The cardinality of a deletion set F can be compared both to $|F|$, but also to $|w|$, for various strings w such that $w \rightarrow L = F$ for some language L. For instance, we can have $|w| = |F|$ (when $\lambda \in L, w \in F$). In such a case, $card(F)$ is smaller than $M(F)$.

Theorem 2. *If $F = w \rightarrow L, |F| = |w|$, then $card(F) \leq M(F) - |F|$.*

Proof. Every string of length $0, 1, \ldots, |F| - 1$ in F corresponds to a substring of length $|w|, |w| - 1, \ldots, 1$, respectively, in w and there are $1, 2, \ldots, |w|$ such substrings. Adding the string of length $|F| = |w|$ (it can be obtained by deleting λ from w), we obtain $\frac{|w|(|w|+1)}{2} + 1$ possibilities, which is equal to $M(F) - |F|$. ♣

The upper bound in Theorem 2 can be reached if and only if the string w consists of distinct symbols. The *if* part is obvious (all strings w_1, w_2, for $w = w_1 v w_2$, $v \in V^*$, are distinct). The converse implication follows from the next result.

Theorem 3. *If $F = w \rightarrow L, w = w_1 c w_2 c w_3, c \in V, w_1, w_2, w_3 \in V^*$, then*

$$card(F) < \frac{|w|(|w|+1)}{2} + 1,$$

for all $L \subseteq V^*$.

Proof. The value $M(F) - |F|$ in the previous theorem is reached when for all $t, 0 \leq t \leq |w| - 1$, we have

$$card((w \rightarrow L) \cap V^t) = t + 1.$$

However, for a string w as above we have

$$w_1 c w_3 \in (w \rightarrow w_2 c) \cap (w \rightarrow c w_2),$$

i.e. for $t = |w_1w_2| + 1$ we get at most t strings of length t, hence the inequality in the theorem is proper. ♣

Corollary. *If $w \in V^*, |w| > card(V)$, then $card(w \rightarrow L) < \frac{|w|(|w|+1)}{2} + 1$, for all $L \subseteq V^*$.*

The next relations are obvious.

Lemma 2. *(i) For all $w \in V^*, L \subseteq V^*$, we have*

$$w \rightarrow L = w \rightarrow (L \cap Sub(w)).$$

(ii) For all $w \in V^*, L_1 \subseteq L_2 \subseteq V^*$, we have

$$card(w \rightarrow L_1) \leq card(w \rightarrow L_2).$$

Therefore,

$$card(w \rightarrow V^*) = max\{card(w \rightarrow L) \mid L \subseteq V^*\}.$$

These remarks naturally raise the following *problem.* Denote, for given $w \in V^*$,

$$d(w) = card(w \rightarrow V^*),$$

and define

$$Ef(V) = \{x \in V^* \mid d(x) \geq d(y) \text{ for all } y \in V^*, |y| = |x|\}$$

(the most efficient strings in V^*, namely the strings which lead to deletion sets of maximal cardinality, compared with other strings of the same length).

Problems: characterize this language; which is its place in the Chomsky hierarchy ?

Surprisingly enough, the language $Ef(V)$ does not seem to be "too complex". More specifically, we have

Theorem 4. *Consider $V = \{a_1, a_2, \ldots, a_s\}$ with $s \geq 2$, and $w \in V^*, |w| = n$. Then*

$$d(w) \leq \frac{n^2 + 2n + 2}{2} - \frac{1}{2}\sum_{i=1}^{s}(|w|_{a_i})^2.$$

This bound is maximal when

$$-1 \leq |w|_{a_i} - |w|_{a_j} \leq 1,$$

for all $1 \leq i, j \leq s$, and this value can be reached.

Proof. If all symbols in w are distinct, then we have $d(w)$ as given by Theorem 2, $d(w) = \frac{n(n+1)}{2} + 1$.

For every $i, 1 \leq i \leq s$, such that $|w|_{a_i} \geq 2$ and for each pair of occurrences of a_i in w, $w = w_1a_iw_2a_iw_3$, both $w \rightarrow a_iw_2$ and $w \rightarrow w_2a_i$ contain the string $w_1a_iw_3$. Thus, it is enough to count the pairs of occurrences of a_i in w, for all $i, 1 \leq i \leq s$.

Denote $|w|_{a_i} = k_i, 1 \leq i \leq s$.

Clearly, there are $\frac{k_i(k_i-1)}{2}$ pairs of occurrences of the symbol a_i in w. Therefore we obtain

$$d(w) \leq \frac{n(n+1)}{2} + 1 - \sum_{i=1}^{s} \frac{k_i(k_i-1)}{2} = \\ = \frac{n^2+n+2}{2} - \frac{1}{2}\left(\sum_{i=1}^{s} k_i^2 - \sum_{i=1}^{s} k_i\right).$$

Replacing $\sum_{i=1}^{s} k_i$ by n in this expression we get a bound for $d(w)$ as in the theorem.

Consider now strings w of the form

$$w = a_1^{k_1} a_2^{k_2} \ldots a_s^{k_s},$$

and evaluate the cardinality of $w \to V^*$. Deleting any one of the k_i occurrences of the symbol a_i we obtain exactly one string in $w \to V^*$, hence k_i possible choices can produce only one string. Count a "loss" of $k_i - 1$ strings. In general, deleting j occurrences of a_i we get one string although we have $k_i - j + 1$ possibile choices of the j occurrences, hence we have a "loss" of $k_i - j$ strings. If we delete from w strings v containing occurrences of two distinct symbols a_i, a_j, then all the obtained strings are distinct. In conclusion, we lose exactly

$$\sum_{i=1}^{s}((k_i-1) + (k_i-2) + \ldots + 1) = \sum_{i=1}^{s} \frac{k_i(k_i-1)}{2}$$

strings. Therefore, for such strings w we obtain

$$d(w) = \frac{n(n+1)}{2} + 1 - \sum_{i=1}^{s} \frac{k_i(k_i-1)}{2} = \\ = \frac{n^2+2n+2}{2} - \frac{1}{2}\sum_{i=1}^{s} k_i^2.$$

Now, clearly, this value is maximal when $\sum_{i=1}^{s} k_i^2$ is minimal. Because $\sum_{i=1}^{s} k_i$ is constant, this happens when the values of k_i are as close as possible to n/s (when $n = ks$, then $k_i = k$ for all $1 \leq i \leq s$; when $n = ks + s', s' < s$, then s' symbols appear $k+1$ times in w and the other $s - s'$ symbols appear k times). ♣

For example, in the case $s = 2$ we get

$$d(w) \leq \begin{cases} (m+1)^2, & \text{if } |w| = 2m, \\ (m+1)(m+2), & \text{if } |w| = 2m+1, \end{cases}$$

and this value is maximal and reached for strings $a^m b^m, b^m a^m$, respectively for strings $a^m b^{m+1}, a^{m+1} b^m, b^m a^{m+1}, b^{m+1} a^m$.

Corollary 1. *For any $n \geq 0$, if $n \equiv r \mod s$, where $s = card(V)$, then*

$$max\{d(w) \mid |w| = n\} = n + 1 + \frac{(s-1)n^2 - sr + r^2}{2s}.$$

Proof. It follows from the proof of the theorem that, if $n = ks + r$, then the largest value of $d(w)$, where $|w| = n$, is reached when $|w|_{a_i} = k + 1$, $1 \leq i \leq r$, and $|w|_{a_i} = k$ for $r \leq i \leq s$. The resulting value of $d(w)$ is the expression in the corollary, and this value equals the bound, i.e. the maximal value of $\frac{n^2+2n+2}{2} - \frac{1}{2}\sum_{i=1}^{s}(|w|_{a_i})^2$. ♣

Corollary 2. *For V containing at least three symbols, $Ef(V)$ is not context-free; if V contains two symbols, then $Ef(V)$ is not regular.*

Proof. Take an order of symbols in V, $V = \{a_1, a_2, \ldots, a_n\}, n \geq 3$; then, according to the theorem,

$$Ef(V) \cap a_1^* a_2^* \ldots a_n^* = \{a_1^{i_1} a_2^{i_2} \ldots a_n^{i_n} \mid -1 \leq i_j - i_k \leq 1 \text{ for all } 1 \leq j, k \leq n, i_j \geq 0, 1 \leq j \leq n\},$$

and this language is not context-free.

In the same way one can see that $Ef(\{a, b\})$ is not regular. ♣

We hope to return to a more detalied study of the languages $Ef(V)$.

We discuss here some further (significant) examples. Take

$$F = \{aba, abba\} = abba \rightarrow \{\lambda, b\},$$

and $z = b^5$. Then $F \cup \{z\}$ is not a deletion set. (If $w \rightarrow L = F \cup \{z\}$, then from $aba, abba \in w \rightarrow L$ we infer that $w = aw'a$ and then we cannot have $b^5 \in w \rightarrow L$).

In this example, $|z| > |F|$. A similar result (a set which is not a deletion set) can be obtained by adding to the previous F a string shorter than $|F|$: such a string is bab.

Call a language $L \subseteq V^*$ *totally prefixed* (*totally suffixed*) if for all $u, v \in L$ we have $u \in Pref(v)$ or $v \in Pref(u)$ (respectively, $u \in Suf(v)$ or $v \in Suf(u)$).

Theorem 5. *Every totally prefixed or totally suffixed finite language F is a deletion set.*

Proof. If F is totally prefixed and finite, then there is $z \in F$ such that $F \subseteq Pref(z)$. Take $w = z\$$, where $\$$ is a new symbol, and define

$$L = \{v \in V^* \mid z = uv \text{ for some } u \in F\}.$$

Then $F = w \rightarrow L\$$. The argument for a totally suffixed F is analogous. ♣

Using this remark we can find many deletion sets starting from DOL languages. For instance, if $G = (V, h, u)$ is a DOL system such that

$$h(u) = uv,$$

then $L(G)$ is a totally prefixed language, hence any finite subset of $L(G)$ is a deletion set. This is the case with the DOL system

$$G = (\{a, b\}, h, a),$$

corresponding to the Thue morphism $h(a) = ab, h(b) = ba$.

Similarly, the system

$$G' = (\{a, b\}, h', b),$$

with $h(a) = b, h(b) = ab$, generates a totally suffixed language. Even taking the axiom a instead of b, that is considering the Fibonacci sequence [3]

$$a, b, ab, bab, abbab, bababbab, \ldots$$

still each finite subset of this DOL language is a deletion set. Indeed, denoting by $x_i, i \geq 1$, the strings in this sequence, for $i \geq 2$ we have

$$x_{i+2} = x_i x_{i+1},$$

hence every finite $F \subseteq \{x_2, x_3, \ldots\}$ is a deletion set. If we have $F \subseteq \{x_1, x_2, \ldots\}$ and $x_1 \in F$, then we take $x_n \in F$ with maximal n and construct

$$\begin{aligned} w &= a\$x_n, \\ L &= \{\$x_n\} \cup \{a\$v \mid x_n = vu, \text{ for some } u \in F\}. \end{aligned}$$

We obtain $F = w \rightarrow L$.

Say that a DOL system G has the *deletion property* iff every finite subset of $L(G)$ is a deletion set.

It is not surprising that there are DOL systems which do not have this property. For example, consider

$$G = (\{a, b\}, h, bab),$$

with $h(a) = a, h(b) = bb$. We obtain the sequence

$$bab, b^2ab^2, b^4ab^4, \ldots, b^{2^i}ab^{2^i}, \ldots$$

No set $F \subseteq L(G)$, containing at least three strings is a deletion set. (Indeed, if $b^jab^j \in w \rightarrow L$, then either $b^ja \in Pref(w)$, or $ab^j \in Suf(w)$; for two strings b^jab^j, b^kab^k, one will fix the prefix of w and the other will fix the suffix; a third string cannot then be obtained from w by deletion).

Thus we are led to the following natural *problems*: Characterize the DOL systems having the deletion property. Is it decidable whether or not an arbitrarily given DOL system has the deletion property ?

We close this section by pointing out that every finite language over an one-letter alphabet is a deletion set.

4. Deciding whether a Set is a Deletion Set

The main result of this section is the next one.

Theorem 6. *It is decidable whether a given finite set is a deletion set.*

Proof. Take $F \subseteq V^*, |F| = m$.

If $card(V) = 1$, then F is a deletion set; therefore, assume $card(V) \geq 2$ and take two symbols $a, b \in V, a \neq b$.

Claim 1. *Given a string w and a set F, it is decidable whether or not there exists F' such that $w \rightarrow F' = F$.*

Indeed, it is enough to consider all finite sets F' with $|F'| \leq |w|$. Their number is finite and for each of them we can check whether $w \rightarrow F' = F$ or not.

Claim 2. *Assume F with $|F| = m$ is a deletion set. Then there is w with $|w| \leq 3m+3$ and F' such that $w \rightarrow F' = F$.*

Indeed, assume $F = w' \rightarrow F''$ for some w' with $|w'| > 3m+3$. We can write $w' = w_1 w_3 w_2$ with $|w_1| = |w_2| = m$. Every word of F is obtained by concatenating a prefix of w_1 with a suffix of w_2, hence w_3 is useful only for guiding, together with F'', which prefix and which suffix are taken. We are free to choose F''. Thus, defining

$$w = w_1 b a^{m+1} b w_2,$$
$$F' = \{v_1 b a^{m+1} b v_2 \mid \text{there is } z \in F, z = u_1 u_2, \text{such that } u_1 v_1 b a^{m+1} b v_2 u_2 = w\},$$

we obtain $F = w \rightarrow F'$.

The equality is obvious, because there is no substring of the form a^{m+1} in w_1 or in w_2.

Now, combining Claims 1 and 2 we obtain the theorem. ♣

The construction of the string w and of the set F' in the proof of Claim 2 raises the question whether or not the length of w can be decreased. More specifically, on the one hand it is natural to ask whether or not the substring $ba^{m+1}b$ separating w_1, w_2 is necessary and, on the other hand – in the affirmative case – whether or not a shorter string can be used.

The separating string is necessary, as the next *example* shows: consider

$$F = \{c, ca, cab, caba, cabab, cc, acc, bacc, abacc, cabcc\}.$$

We have $|F| = 5$, and the only possibility is to take

$$w_1 = cabab, \ w_2 = abacc.$$

However, we cannot use $w = w_1 w_2$, because, in order to get $cabcc$, we must have $ababa \in F'$, and

$$cabababacc \rightarrow ababa = \{cbacc, cabcc\},$$

which implies $cbacc \in F$, a contradiction.

Thus, it is of interest to ask whether a string shorter than $ba^{m+1}b$ can separate w_1 and w_2 in the previous proof. In general, this is the case (thus a speed-up of the algorithm suggested by the proof is obtained). For instance, we can take as w_3 any string in the set

$$V^k - Sub(w_1 w_2),$$

for the smallest k for which this set is non-empty. We have

$$V^k - Sub(w_1 w_2) = V^k - (Sub(w_1 w_2) \cap V^k),$$

and

$$card(V^k) = (card(V))^k,$$
$$Sub(w_1 w_2) \cap V^k \leq 2m - k + 1,$$

(remember that $|w_1w_2| = 2m$). Therefore, $|w_3| \leq k$, for the smallest k such that

$$(card(V))^k > 2m - k + 1.$$

For example, for $V = \{a, b\}$, we have to compare 2^k with $2m - k + 1$. Here are some values of k, for small m:

m	1	2	3	4	5	6	7	8	9	10
minimal k	2	2	3	3	4	4	4	4	4	5

The improvement is significant, in comparison with the case of $|w_3| = m + 3$ in the proof of Theorem 6.

5. A Characterization of Deletion Sets

We present a simple, combinatorial characterization, having a series of interesting consequences.

Theorem 7. *$F \subseteq V^*$ is a deletion set if and only if there is $z \in V^*$ such that $F \subseteq Pref(z)Sub(z)$.*

Proof. If F is a deletion set, $F = w \rightarrow F'$, then we have $F \subseteq Pref(w)Suf(w)$ (if $x \in F$, then $x = x_1x_2$ for $w = x_1yx_2, y \in F'$, hence $x \in Pref(w)Suf(w)$).

Conversely, take $F \subseteq Pref(z)Suf(z)$ and consider

$$\begin{aligned} w &= z\$z, \\ F' &= \{y_1\$y_2 \mid \text{ there is } x \in F, x = x_1x_2, x_1 \in Pref(z), \\ &\qquad x_2 \in Suf(z), \text{ such that } z = x_1y_1 = y_2x_2\}, \end{aligned}$$

where \$ is a new symbol. The equality $F = w \rightarrow F'$ is obvious. ♣

Remark 1. We sometimes use a new symbol (such as \$ here) as marker. Such an extension of the alphabet V is not necessary, provided V contains at least two symbols a and b. Then the marker can be replaced by a word ba^ib, where i is sufficiently large. The minimization of such a separator word is often a nontrivial task. Observe that we could have replaced $ba^{m+1}b$ in Claim 2 in the preceding section by a marker, and that we also considered there the minimization problem. Thus, markers are by no means essential; we use them only to facilitate the reading.

Corollary 1. *(i) If F is a deletion set, then every subset of F is a deletion set.*
(ii) If F_1, F_2 are deletion sets with $F_i \subseteq Pref(z_i)Suf(z_i), i = 1, 2$, and $z_1 = z_2$, then $F_1 \cup F_2$ is a deletion set.

Corollary 2. *If F is a deletion set, then xF, Fy, xFy are deletion sets for all strings x, y.*

Proof. Since F is a deletion set, there is z such that $F \subseteq Pref(z)Suf(z)$. Then

$$xF \subseteq Pref(xz)Suf(xz), \quad (1)$$
$$Fy \subseteq Pref(zy)Suf(zy), \quad (2)$$
$$xFy \subseteq Pref(xzy)Suf(xzy), \quad (3)$$

therefore xF, Fy, xFy are deletion sets.

We prove only (3). Take $w \in F$, hence $xwy \in xFy$.

From $F \subseteq Pref(z)Suf(z)$ we obtain $w = w_1w_2$ for $z = w_1u_1 = u_2w_2$. Then $xwy = xw_1w_2y$, and

- $xw_1u_1 = xz$, hence $xw_1 \in Pref(xz)$,
- $u_2w_2y = zy$, hence $w_2y \in Suf(zy)$.

Therefore $xwy \in Pref(xz)Suf(zy) \subseteq Pref(xzy)Suf(xzy)$. ♣

Application. $F = \{a^3, b^3, aba\}$ is not a deletion set.

Indeed, assume $F \subseteq Pref(z)Suf(z)$ for some $z \in \{a,b\}^*$. From $a^3 \in F$ it follows that z either begins or ends by a. Assume z begins by a; the other case is similar. Then $b^3 \in Suf(z)$, hence z ends by b. This implies $aba \in Pref(z)$.

However, $aba \in Pref(z), a^3 \in Pref(z)$ imply $a = b$, a contradiction.

6. An Algebraic Approach

We shall now give another characterization of deletion sets as well as a new algorithm for deciding whether or not a set is a deletion set.

For this purpose, the following order relation over $V^* \times V^*$ is used: for (u_1, u_2), $(v_1, v_2) \in V^* \times V^*$ we write

$$(u_1, u_2) \leq (v_1, v_2) \text{ iff } (u_1, u_2) = (\alpha, \gamma\delta) \text{ and } (v_1, v_2) = (\alpha\beta, \delta),$$

for some words $\alpha, \beta, \gamma, \delta \in V^*$ (some of them may be λ).

Hence, for any $\alpha, \beta, \gamma, \delta \in V^*, (\alpha, \gamma\delta) \leq (\alpha\beta, \delta)$, and conversely, if $(u_1, u_2) \leq (v_1, v_2)$, then there exist $\alpha, \beta, \gamma, \delta$ such that $(u_1, u_2) = (\alpha, \gamma\delta)$ and $(v_1, v_2) = (\alpha\beta, \delta)$.

Lemma 3. *The relation "$\leq$" is a partial order relation over $V^* \times V^*$.*

Proof. If $(u_1, u_2) \leq (v_1, v_2)$ and $(v_1, v_2) \leq (u_1, u_2)$, then u_1 is a prefix of v_1 and v_1 is a prefix of u_1, hence $u_1 = v_1$. Moreover, v_2 is a suffix of u_2 and u_2 is a suffix of v_2. Therefore, $u_2 = v_2$ and consequently "$\leq$" is antisymmetric.

If $(u_1, u_2) \leq (v_1, v_2)$ and $(v_1, v_2) \leq (w_1, w_2)$, then u_1 is a prefix of w_1 and w_2 is a suffix of u_2. Therefore $(u_1, u_2) \leq (w_1, w_2)$ and the relation "$\leq$" is transitive. ♣

Let w be in $V^*, w = a_1a_2 \ldots a_n, a_i \in V, 1 \leq i \leq n$. Denote by P_w the following set of pairs:

$$P_w = \{(a_1a_2 \ldots a_i, a_ja_{j+1} \ldots a_n) \mid i \leq j\} \cup$$
$$\cup \{(\lambda, a_ja_{j+1} \ldots a_n) \mid 1 \leq j\} \cup \{(a_1a_2 \ldots a_i, \lambda) \mid i \leq n\}.$$

Lemma 4. *For any $w \in V^*, P_w$ is a lattice with the first and last element, namely $(\lambda, a_1a_2 \ldots a_n)$ and $(a_1a_2 \ldots a_n, \lambda)$, respectively.*

Proof. Let $x = (a_1a_2 \ldots a_i, a_ja_{j+1} \ldots a_n), y = (a_1a_2 \ldots a_k, a_ra_{r+1} \ldots a_n)$ be in P_w and define $p_1 = min(i,k),\ p_2 = max(i,k),\ q_1 = min(r,j),\ q_2 = max(r,j)$. Now, let us consider

$$m = (a_1a_2 \ldots a_{p_1}, a_{q_1} \ldots a_n), \text{ and } M = (a_1a_2 \ldots a_{p_2}, a_{q_2} \ldots a_n).$$

It is not difficult to prove that

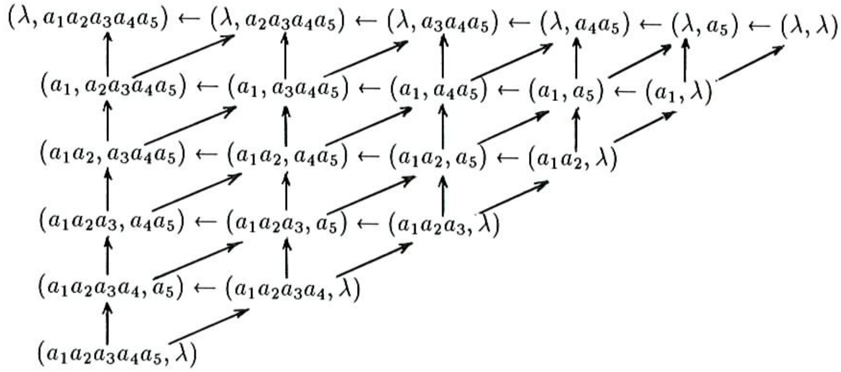
$$m = inf\{x, y\} \text{ and } M = sup\{x, y\}.$$

Moreover, m and M are in P_w.

Obviously, $(\lambda, a_1a_2 \ldots a_n)$ is the first element and $(a_1a_2 \ldots a_n, \lambda)$ is the last element of the lattice. ♣

Remark 2. Assume that $w \to F' = F$. Then every word $z \in F$ has a decomposition $z = z_1z_2$ such that the pair (z_1, z_2) is in P_w.

For illustration, consider the Hasse diagram of the lattice P_w, for $w = a_1a_2a_3a_4a_5$. An arrow $x \leftarrow y$ means that $x \leq y$.



If $F = \{u_1, u_2, \ldots, u_m\}$ is a set of words over V^*, then we call *a decomposition set* of F a set of the following form

$$P_F = \{(u'_1, u''_1), \ldots, (u'_m, u''_m)\},$$

where $u'_iu''_i = u_i, 1 \leq i \leq m$.

If $w \to F' = F$, then there exists a decomposition set of F, P_F, such that $P_F \subseteq P_w$. Note that for a given finite set F, there are only a finite number of sets P_F.

Let P_F be such a set. We may try to complete P_F to a lattice using the following algorithm:

Algorithm for completion to a lattice:

- *Input.* P_F
- *Step 1.* If P_F is a lattice, then *accept* P_F and STOP.
- *Step 2.* Let $x = (x', x''), y = (y', y'')$ be in P_F such that $inf\{x, y\}$ and/or $sup\{x, y\}$ is not in P_F.

  If $x' \notin Pref(y')$ and $y' \notin Pref(x')$, or $x'' \notin Suf(y'')$ and $y'' \notin Suf(x'')$, then *reject* P_F and STOP, else compute $z = inf\{x, y\}, t = sup\{x, y\}$, $P_F = P_F \cup \{z, t\}$, goto *Step 1.*

Note that the computation of $z = inf\{x, y\}$ can be easily done: $z = (z_1, z_2)$, where

$$z_1 = \begin{cases} x', & \text{if } y' \in Pref(x'), \\ y', & \text{if } x' \in Pref(y'), \end{cases}$$

$$z_2 = \begin{cases} x'', & \text{if } y'' \in Suf(x''), \\ y'', & \text{if } x'' \in Suf(y''), \end{cases}$$

and similarly, $t = (t_1, t_2)$, where

$$t_1 = \begin{cases} x', & \text{if } y' \in Pref(x'), \\ y', & \text{if } x' \in Pref(y'), \end{cases}$$

$$t_2 = \begin{cases} x'', & \text{if } x'' \in Suf(y''), \\ y'', & \text{if } y'' \in Suf(x''). \end{cases}$$

The above algorithm does terminate because the new pairs z, t that are added to P_F have the property that $|z_1| + |z_2|$, $|t_1| + |t_2|$ do not exceed the constant $c = c_1 + c_2$, where

$$c_1 = max\{|\alpha_i'| \mid (\alpha_i', \alpha_i'') \in P_F, 1 \leq i \leq m\},$$
$$c_2 = max\{|\alpha_i''| \mid (\alpha_i', \alpha_i'') \in P_F, 1 \leq i \leq m\}.$$

Here P_F and m refer to the originally given items. Thus, the algorithm will consider only a finite number of new pairs.

Theorem 8. *A set F is a deletion set if and only if F has a decomposition set Q_F such that Q_F can be completed to a lattice P_F (with the first and last element).*

Moreover, if $m = (m_1, m_2)$ is the first element of P_F and $M = (M_1, M_2)$ is the last element of P_F, then we can define $w = M_1 \# m_2$, where $\#$ is a new symbol, and find F' by appropriate definition, such that $w \to F' = F$.

Proof. ($\Rightarrow$) Assume that F is a deletion set, i.e. there are w and F' such that $w \to F' = F$. Define the decomposition set

$$Q_F = \{(\alpha', \alpha'') \mid \alpha'\alpha'' \in F, \text{ there is } \beta \in F', \text{ such that } w = \alpha'\beta\alpha''\}.$$

Now, assume that $u = (u_1, u_2), v = (v_1, v_2)$ are in Q_F and define

$$z_1 = \begin{cases} u_1, & \text{if } u_1 \in Pref(v_1), \\ v_1, & \text{if } v_1 \in Pref(u_1), \end{cases}$$

$$z_2 = \begin{cases} u_2, & \text{if } v_2 \in Suf(u_2), \\ v_2, & \text{if } u_2 \in Suf(v_2), \end{cases}$$

$$t_1 = \begin{cases} u_1, & \text{if } v_1 \in Pref(u_1), \\ v_1, & \text{if } u_1 \in Pref(v_1), \end{cases}$$

$$t_2 = \begin{cases} u_2, & \text{if } u_2 \in Suf(v_2), \\ v_2, & \text{if } v_2 \in Suf(u_2), \end{cases}$$

$z = (z_1, z_2)$ and $t = (t_1, t_2)$.

It is easy to prove that $z = inf\{u, v\}$ and $t = sup\{u, v\}$. Thus, Q_F can be completed with z and t. This procedure can be repeated until Q_F is completed to a lattice, P_F.

The first element of P_F is $m = (m_1, m_2)$, where m_1 is the shortest prefix of w which is not deleted by any $x, x \in F'$, and m_2 is the longest suffix of w which is not deleted by any $y, y \in F'$.

Similarly, the last element of P_F is $M = (M_1, M_2)$, where M_1 is the longest prefix of w which is not deleted by any $x, x \in F'$, and M_2 is the shortest suffix of w which is not deleted by any $y, y \in F'$.

($\Leftarrow$) Assume that P_F is the lattice computed by the above algorithm, starting from a decomposition set Q_F of F.

Define w as $w = M_1\#m_2$, where $\#$ is a new symbol and $M = (M_1, M_2)$, $m = (m_1, m_2)$ is the last, respectively the first element of P_F.

For any $u = (u', u'') \in P_F$, u' is a prefix of M_1 and u'' is a suffix of m_2, i.e. there are two strings α and β in V^* such that $M_1 = u'\alpha$ and $m_2 = \beta u''$. Then, the word $\alpha\#\beta$ is, by definition, an element of F'. Therefore,

$$F' = \{\alpha\#\beta \mid \text{there is } (u', u'') \in P_F, \text{ such that } u'\alpha = M_1, \beta u'' = m_2\}.$$

Now, it is easy to observe that

$$M_1\#m_2 \to F' = F.$$

Thus $w \to F' = F$ and therefore F is a deletion set. ♣

Note that P_F is the sublattice generated by Q_F, i.e. P_F is the smallest lattice of $V^* \times V^*$ which contains Q_F (see [1] for terminology).

Remark 3. As regard the marker #, we refer to Remark 1. The new symbol # is necessary, as we can see from the following example. Let P_F be as in the next diagram:

$$\begin{array}{ccc} (ab, bd) & \longleftarrow & (ab, \lambda) \\ \uparrow & & \uparrow \\ (abc, d) & \leftarrow - - - & (abc, \lambda) \end{array}$$

The dotted part of the diagram is completed by the previous algorithm.

Now, $m = (ab, bd)$ and $M = (abc, \lambda)$, but we cannot define $w = abcbd$ because then $b \in F$ and $w \to b = \{acbd, abcd\}$. But $acbd \notin F$.

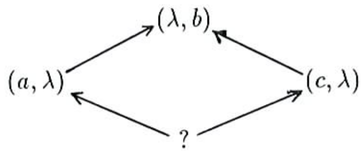
On the other hand, if we define $w = abc\#bd$, then the definition of F' is possible, too: $F' = \{c\#bd, c\#, \#b\}$. Note that

$$abc\#bd \to \{c\#bd, c\#, \#b\} = \{ab, ab^2d, abcd\}.$$

It is likely that the above method leads to a word w which is "almost" the shortest possible. However, the algorithm needs a lot of computation, because for a fixed set F the number of decomposition sets P_F is large.

We close this section by considering one more example. Take $F = \{a, b, c\}$, which we have pointed out in Section 3 is not a deletion set. Each decomposition set of F has pairs of the form (x, y), where $x = \lambda$ and $y \neq \lambda$ or $y = \lambda$ and $x \neq \lambda$.

No such decomposition set can be completed to a lattice. For example, if $Q_F = \{(a, \lambda), (\lambda, b), (c, \lambda)\}$, then we obtain



The value of $sup\{(a,\lambda),(c,\lambda)\}$ is undefined.

7. Undecidability Results

In contrast to Theorem 6, it is proved in [2] (Theorem 5.41) that given a set $F = \{a, b\}$ and an arbitrary context-free language, L, it is undecidable whether or not a string w exists such that $F = w \to L$. We shall prove here such a result for any deletion set F, different from $\{\lambda\}$.

In order to fix the terminology, we say that a deletion set F is *context-free decidable* (CF-decidable, for short) if the problem "given $L \in CF$, does there exist w such that $F = w \to L$?" is decidable. (Similarly, we can say that a language F is *REG-decidable* when given $L \in REG$ we can decide whether or not a string w exists such that $w \to L = F$.)

The next theorem is somewhat unexpected – at least in the case of deletion sets of cardinality one and in comparison to Theorem 5.36 in [2] which says that any finite set is REG-decidable.

Theorem 9. *No deletion set F different from $\{\lambda\}$ is CF-decidable.*

Proof. Case 1. Assume $F = \{z\}, z \neq \lambda$.
Take an arbitrary context-free language L_0 and construct

$$L = \#V^*\# \cup \bigcup_{uv=z} u\#L_0\#v.$$

Then, there is w such that $w \to L = F$ if and only if $L_0 \neq V^*$ (which is not decidable for arbitrary context-free languages).

Indeed, if $L_0 \subset V^*$, then, for $x \in V^* - L_0$ we take $w = z\#x\#$ and we have $w \to L = \{z\}$.

If $L_0 = V^*$, then suppose that there is w such that $w \to L = \{z\}$. We distinguish two cases:

(1) $z \in w \to \#V^*\#$. Then $w = u\#x\#v$ for some $x \in V^*, uv = z$. But $x \in L_0$, hence $w \to u\#L_0\#v = \{\lambda\} \subseteq w \to L = F$, a contradiction.

(2) $z \in w \to u\#L_0\#v$, for some $u, v \in V^*, uv = z$. Then $w = u'u\#x\#vv'$, for some $x \in L_0, u'v' = z$. But $x \in V^*$, hence $w \to \#V^*\# = \{u'uvv'\} \subseteq w \to L = F$ and $|u'uvv'| = 2|z|$, a contradiction.

Case 2. Assume $card(F) = 2$ and $F = \{\lambda, z\}, z \neq \lambda$.
For arbitrary context-free languages L_1, L_2, construct

$$L = \#L_1\#z \cup \#L_2\#.$$

Then there is w such that $w \to L = \{\lambda, z\}$, if and only if $L_1 \cap L_2 \neq \emptyset$ (which is not decidable for arbitrary context-free languages).

If $L_1 \cap L_2 \neq \emptyset$, then take $x \in L_1 \cap L_2$ and consider $w = \#x\#z$. Clearly, $w \rightarrow L = \{\lambda, z\}$.

Assume now that $w \rightarrow L = \{\lambda, z\}$ for some w. We must have $w = u\#x\#v$. No one of $\#L_1\#z$ and $\#L_2\#$ can delete u, hence $u = \lambda$ (in order to obtain λ).

Now, if $\lambda \in w \rightarrow \#L_2\#$, then $v = \lambda$ and then $F = \{\lambda\}$, a contradiction. Therefore, $\lambda \in w \rightarrow \#L_1\#z$, that is $w = \#x\#z$, for some $x \in L_1$.

However, then $w \rightarrow \#L_1\#z = \{\lambda\}$, hence $z \in w \rightarrow \#L_1\#$. This implies $x \in L_2$, too, hence $L_1 \cap L_2 \neq \emptyset$.

Case 3. Assume $card(F \cap V^+) \geq 2$.

We know that F is a deletion set. Let w_0 be a string such that $w_0 \rightarrow F_0 = F$, for some language F_0.

Take an arbitrary context-free language L_0 and construct

$$L = \#L_0\# \cup \{xv\#V^*\#ux \mid uv = z \in F, uxv = w_0, x \in F_0\}.$$

Then, there is w such that $w \rightarrow L = F$ if and only if $L_0 \neq V^*$ (which is not decidable).

If $L_0 \subset V^*$, then take $\alpha \in V^* - L_0$ and consider $w = w_0\#\alpha\#w_0$. Clearly, $w \rightarrow \#L_0\# = \emptyset$. On the other hand,

$$\begin{aligned} w_0\#\alpha\#w_0 \rightarrow xv\#V^*\#ux &= w_0\#\alpha\#w_0 \rightarrow xv\#\alpha\#ux = \\ &= \{u'v'\} \text{ such that } w_0 = u'xv = uxv'. \end{aligned}$$

Since $uxv = w_0$ we obtain $u' = u, v' = v$ and $u'v' \in F$ (from the definition of L). Consequently, $w \rightarrow L \subseteq F$.

The converse inclusion follows in the same way, hence $w \rightarrow L = F$.

Assume now that $L_0 = V^*$ and suppose that there is w such that $w \rightarrow L = F$.

Clearly, w is of the form $y_1\#\alpha\#y_2$, hence $w \rightarrow \#L_0\# = y_1y_2$.

Because F contains at least two non-empty strings, there is $xv\#\alpha\#ux \in L$ such that $w \rightarrow xv\#\alpha\#ux = \{z\} \neq \{\lambda\}$. This implies $w = u'xv\#\alpha\#uxv'$, with $u'v' = z$. However,

$$\begin{aligned} u'xv\#\alpha\#uxv' \rightarrow \#L_0\# &= u'xv\#\alpha\#uxv' \rightarrow \#V^*\# \\ &= u'xv\#\alpha\#uxv' \rightarrow \#\alpha\# \\ &= \{u'xvuxv'\} \in F. \end{aligned}$$

But $uxv = w_0$ (from the construction of L) and $u'v' = z \neq \lambda$, hence $|u'xvuxv'| > |w_0| \geq |F|$, a contradiction which concludes the proof. ♣

In conclusion, no deletion set different from $\{\lambda\}$ is CF-decidable. On the other hand, this particular deletion set, $\{\lambda\}$, has the property of being *CF-universal*, in the sense that for any given context-free language L there is w such that $w \rightarrow L = \{\lambda\}$: take w one of the shortest strings in L. Moreover, we have

Theorem 10. *The set $\{\lambda\}$ is the only CF-universal deletion set.*

Proof. The assertion follows from the next two claims.

Claim 1. Every CF-universal deletion set contains the string λ.

Indeed, assume F is CF-universal and take $x \in F, x \neq \lambda$. Consider the (regular) language

$$L = (Pref(x))^* x (Suf(x))^*.$$

If a string w there is such that $w \to L = F$, then from $x \in w \to L$ we obtain $w = uyv, uv = x, y \in L$. As $u \in Pref(x)$, $v \in Suf(x)$, it follows that $uyv \in L$, that is $w \in L$, which implies that $\lambda \in w \to L = F$.

Claim 2. A set F containing both λ and $x, x \neq \lambda$, cannot be a CF-universal deletion set.

Indeed, take $L = \{x\}$ and assume that there is w such that $w \to L = F$. Then $x \in F$ implies $x \in w \to L$, hence $|w| = 2|x|$; on the other hand, $\lambda \in F$ implies $\lambda \in w \to L$, hence $|w| = |x|$. Consequently, $|x| = 0$, a contradiction. ♣

Remark 4. The notion of universality can be formulated for any family X of languages instead of CF. Then, Claim 1 deals with REG-universal and Claim 2 deals with SING-universal deletion sets, where $SING$ is the family of singleton languages. In conclusion, the theorem says that $\{\lambda\}$ is the only X-universal deletion set for all families X including REG.

Note. Research supported by the Academy of Finland, grant 11281, and the Alexander von Humboldt Foundation.

References

[1] G. Grätzer, *General Lattice Theory*, Birkhäusser Verlag, Basel, Stuttgart, 1978.
[2] L. Kari, *On Insertion and Deletion in Formal Language Theory*, Ph. D. Thesis, Univ. of Turku, Dept. of Math., 1991.
[3] G. Rozenberg and A. Salomaa, *The Mathematical Theory of L Systems*, Academic Press, London, 1980.
[4] A. Salomaa, *Formal Languages*, Academic Press, New York, London, 1973.