S. A. Greibach

One counter languages and the chevron operation

RAIRO. Informatique théorique, tome 13, n° 2 (1979), p. 189-194

<http://www.numdam.org/item?id=ITA_1979__13_2_189_0>
ONE COUNTER LANGUAGES
AND THE CHEVRON OPERATION (*) (1)

by S. A. GREIBACH (2)
Communiqué par J. BERSTEL

Abstract. — For a language L and new symbols a and b, define the chevron of L as <L> = \{a^nw^b|n \geq 0, w \in L\}. The family of one counter languages is strongly resistant to the chevron operation in the sense that <L> is a one counter language if and only if L is regular.

Résumé. — Soit L un langage défini sur un alphabet ne contenant pas les lettres a et b. Alors, <L> = \{a^nw^b|n \geq 0, w \in L\} appartient à la famille des langages à un compteur ssi et seulement si L est un langage rationnel.

The family of linear context-free languages not only is not closed under concatenation but is strongly resistant to concatenation in the following sense. If L_1 and L_2 are languages over disjoint alphabets, then L_1 L_2 is linear context-free only if either L_1 or L_2 is regular [9]. Goldstine showed that the least full semiAFL (family of languages containing at least one nonempty language and closed under union, homomorphism, inverse homomorphism, and intersection with regular sets) containing the 1-bounded languages has the same property [8], and recently Latteux demonstrated this property for the least full semiAFL containing the two-sided Dyck set on one letter [12]. A similar phenomenon has been observed for other operations. The family of ultralinear languages is strongly resistant to Kleene * in the sense that, for a language L and a new symbol c, (Lc)* is ultralinear if and only if L is regular [7]. The least full semiAFL containing the bounded languages is likewise strongly resistant to Kleene * [8].

We can make this concept more precise. For operations on at least two languages, the definition of "strongly resistant" is obvious.

DÉFINITION: Let Φ be a k-ary operation on languages, k \geq 2 and \mathcal{L} a family of languages. We say that \mathcal{L} is strongly resistant to Φ if, whenever Φ(L_1, \ldots, L_k) is in \mathcal{L} and the languages L_i are over pairwise disjoint alphabets, then there is some j such that L_j is regular.

(1) The research reported in this paper was supported in part by the National Science Foundation under Grant No. MCS 78-04725.
(2) Department of System Science, University of California, Los Angeles.
For unary operations, the direct version of this definition (for $k=1$) would be “too strong” since, e.g., $L^*$ can be regular for $L$ a nonregular language. Thus, the “resistance” is to a marked version of the operation.

**Definition:** Let $S_1$ and $S_2$ be sets of unary operations on languages. They are *adequately associated* with each other if, for $i, j$ in $\{1, 2\}$, $i \neq j$, the following holds. For each language $L$ and operation $\Phi_i$ in $S_i$, there exists an operation $\Phi_j$ in $S_j$ such that $\Phi_i(L)$ can be obtained from $L$ using a finite number of applications of homomorphism, inverse homomorphism and intersection with regular sets and exactly one application of $\Phi_j$.

Thus, if $S_1$ and $S_2$ are adequately associated with each other, and $\Phi_i$ is in $S_1$, $\Phi_i(L)$ can always be expressed as $M_1(\Phi_2(M_2(L)))$ for some $\Phi_2$ in $S_1$ and finite state transductions ($a$-transducer mappings) $M_1$ and $M_2$ [5, 6, 13]. For example, if $S_1$ contains only Kleene *, then it is adequately associated with the set $S_2$ of operations $\phi_c$, $c$ an individual symbol, where $\phi_c(L) = \emptyset$ if $c$ appears in $L$, and $\phi_c(L) = (L \cap c^*)$ otherwise. If $S_1$ is the set of $(1, R)$ homomorphic replications $(1, R, h_1, h_2)$ [where $(1, R, h_1, h_2)(L) = \{ h_1(w)h_2(w^R) \mid w \in L \}$], then we can take $S_2$ as the set of operations $\theta_e$ where $\theta_e(L) = \emptyset$ if $c$ appears in $L$, and $\theta_e(L) = \{ wcw^R \mid w \in L \}$ otherwise.

**Definition:** A family of languages $\mathcal{L}$ is *strongly resistant* to a set of unary operations $S_1$ if $S_1$ is adequately associated to a set $S_2$ of unary operations such that, for $\Phi$ in $S_2$ if $\Phi(L) \neq \emptyset$, then $\Phi(L)$ is in $\mathcal{L}$ if and only if $L$ is regular. If $S_1 = \{ \Phi \}$, we say $\mathcal{L}$ is *strongly resistant* to $\Phi$.

One could use “only if” instead of “if and only if” in the definition above. However, if $\mathcal{L}$ does not contain $\Phi(L)$ for $L$ regular, a better expression would be “$\Phi$ is irrelevant to $\mathcal{L}$”! Strong resistance theorems for unary operations go back to Bar-Hillel, Perles and Shamir, who proved that the family of context-free languages is strongly resistant to $(1, R)$ homomorphic replications [1].

We now turn our attention to the “chevron” operation introduced and studied in [3, 4, 10]. For a language $L$ and symbols $a, b$, we write

$$\langle L, a, b \rangle = \{ a^n wb^n \mid n \geq 0, w \in L \}.$$  

If $a$ and $b$ are symbols not in the alphabet of $L$, then it does not matter which symbols fill the roles of $a$ and $b$. In this case, we write $\langle L \rangle$ for $\langle L, a, b \rangle$ and call this “the” chevron operation in the notation of [12]. Strictly speaking, $S_1$ is the set of operations $\langle L, a, b \rangle$ and $S_2$ the set of operations $\Phi_{a,b}$ where $\Phi_{a,b}(L) = \emptyset$ if $a$ or $b$ appear in $L$, and $\Phi_{a,b}(L) = \langle L, a, b \rangle$ otherwise. We take the liberty of speaking of the chevron operation instead of $S_1$ and use $\langle L \rangle$ for $\langle L, a, b \rangle$ with $a$ and $b$ new symbols.
As a corollary of the result on concatenation cited above, Latteux showed that the least full semi-AFL containing the two-sided Dyck set on one letter is strongly resistant to chevron. We now extend this result to the family of one-counter languages. That is, we show that \( \langle L \rangle \) is a one counter language if and only if \( L \) is regular.

The idea behind the result is simple. In order to match the a’s and b’s in \( L \), a one counter machine \( M \) must increase the counter during the a’s and decrease it during the b’s and keep it “steady” while reading \( w \) in \( L \). Hence, a finite state acceptor can simulate \( M \) on \( w \), and so \( L \) is regular.

First, we give a formal definition of a one counter machine and the language it accepts.

**Definition:** A **one counter machine** is a quintuple \( M = (Q, \Sigma, H, q_0, F) \) where \( Q \) is a finite set of states, \( q_0 \) in \( Q \) is the designated initial state, \( F \subseteq Q \) is the subset of accepting states, \( \Sigma \) is the finite input vocabulary and the transition set \( H \) is a finite subset of \( Q \times (\Sigma \cup \{ e \}) \times \{ 0, 1 \} \times N \times Q \), where \( e \) denotes the empty word and \( N \) is the set of integers, positive, negative and zero. Machine \( M \) is **normalized** if \( H \) is a finite subset of \( Q \times \Sigma \times \{ 0, 1 \} \times \{ -1, 0, 1 \} \times Q \).

**Definition:** An instantaneous description (ID) of one counter machine \( M = (Q, \Sigma, H, q_0, F) \) is a triple \( (q, w, z) \) where \( q \) is in \( Q \), \( w \) is in \( \Sigma^* \) and \( z \) is a nonnegative integer, the size of the counter. If \( (q, aw, z) \) is an ID, \( a \) in \( \Sigma \cup \{ e \} \), and \( (q, a, i, j, p) \) is a transition in \( H \) such that \( i = 0 \) if and only if \( z = 0 \) and that \( z + j \geq 0 \), then we write \( (q, aw, z) \vdash (p, w, z + j) \). If \( I_1, \ldots, I_n \) are ID’s with \( I_1 \vdash I_2 \vdash \ldots \vdash I_n \), we call this a computation and write \( I_1 \triangleright I_n \); we also write \( I_1 \triangleright^* I_1 \). If \( I_1 = (q_0, w, 0) \) and \( I_n = (f, e, 0) \) for some \( f \) in \( F \), then \( I_1 \triangleright I_n \) is an accepting computation for input \( w \). The language accepted by \( M \) is

\[
L(M) = \{ w \in \Sigma^* \mid \text{there is an accepting computation for input } w \}
\]

and is called a **one counter language**.

Thus, a one counter machine \( M \) is a nondeterministic machine with a one-way input tape. It has one register which contains a nonnegative integer. The effect of a transition \( (q, a, i, j, p) \) is that, depending on the current state \( (q) \), input (if \( a \neq e \)), and whether or not the counter is zero (whether \( i = 0 \)), the machine can change state (to \( p \)), add \( j \) to the counter (for \( j \geq 0 \)) or subtract \( |j| \) from the counter (for \( j < 0 \)) and either advance the input tape (\( a \neq e \)) or leave it alone (\( a = e \); this is an \( e \)-move). The machine accepts \( w \) if it can start in the initial state with the counter 0 and reach an accepting state with the input completely scanned and the counter 0.

A one counter machine is normalized if, at one step, it can add or subtract at most 1 and it must advance the input tape at every step. If \( L \) is a one counter
language, then \( L = L(M) \) for some normalized one counter machine [11]. Hence, we can assume without loss of generality that our machines are normalized.

First, we use the familiar counting argument to show that, if \( \langle L \rangle = L(M) \), and \( M \) has \( k \) states, then for each \( m > 0 \) there is an integer \( n, 1 \leq n \leq km + 1 \) such that every accepting computation for input \( a^n wb^n \) must have counter size at least \( m \) throughout the scan of \( w \).

**Lemma 1:** Let \( \langle L \rangle = L(M) \) for a one counter machine \( M \) with \( k \) states. For each \( m > 0 \), there is an integer \( n, 0 < n \leq km + 1 \), such that, for every \( w \) in \( L \) and every accepting computation of \( M \) for \( a^n wb^n \), the counter size does not drop below \( m \) during the processing of \( w \).

**Proof:** Suppose the lemma is false for \( m > 0 \). The argument is the familiar information theoretic one. There are at most \( km \) configurations with counter size below \( m \). However, for each integer \( n, 1 \leq n \leq km + 1 \), there is some \( w \) in \( L \) and some accepting computation for \( a^n wb^n \) which enters a configuration with counter size below \( m \) while reading \( w \). Thus, there must be integers \( n_1 \) and \( n_2 \), \( n_1 \neq n_2 \), words \( w_1 = x_1 y_1 \) and \( w_2 = x_2 y_2 \) in \( L \) and accepting computations \( C_i \) for input \( a^n w_i b^n \), \( i = 1, 2 \) which enter the same configuration after reading \( a^n x_i \). Thus, by splicing together the first part of computation \( C_1 \) and the last part of computation \( C_2 \), we obtain an accepting computation for \( a^n x_1 y_2 b^n \), a contradiction. Hence the lemma must hold. □

Now we use lemma 1 to show that, if \( M \) has \( k \) states and we take \( m = k + 1 \), then the counter cannot increase by more than \( k \) during the scan of \( w \). The proof of lemma 2 uses an idea similar to the one underlying the iteration theorems of [2], which could not be used directly (because [2] uses strict iterative pairs).

**Lemma 2:** Let \( \langle L \rangle = L(M) \) for a normalized one counter machine with \( k \) states. There is an integer \( n, k + 1 \leq n \leq k(k + 1) + 1 \) such that, for every \( w \) in \( L \) and every accepting computation for input \( a^n wb^n \), the counter size does not fall below \( k + 1 \) nor increase by more than \( k \) during the scan of \( w \).

**Proof:** Lemma 1 tells us that there is an integer \( n \leq k(k + 1) + 1 \) such that, for every \( w \) in \( L \) and every accepting computation for input \( a^n wb^n \), the counter size does not fall below \( k + 1 \) during the scan of \( w \). We claim that the counter size also cannot increase by more than \( k \) during the scan of \( w \), for otherwise we could pump up a subword of \( w \) and a subword of \( b^n \) and get an accepting computation for a word not in \( \langle L \rangle \). Note that \( n \geq k + 1 \), since \( M \) is normalized.

Consider an accepting computation \( C \) for input \( a^n wb^n \), \( w \) in \( L \). This computation can be divided into pieces \( C_1, C_2, C_3 \) with

\[
C_1 : (q_0, a^n wb^n, 0) \xrightarrow{\ast} (q_1, wb^n, z_1),
\]

\[
C_2 : (q_1, wb^n, z_1) \xrightarrow{\ast} (q_2, b^n, z_2),
\]

\[
C_3 : (q_2, b^n, z_2) \xrightarrow{\ast} (f, e, 0),
\]

R.A.I.R.O. Informatique théorique/Theoretical Informatics
where \( q_0 \) is the initial state, \( f \) is some accepting state, the counter size is at least \( k + 1 \) through \( C_2 \) and so in particular never becomes 0, and \( z_1, z_2 \geq k + 1 \).

Suppose the counter size increases by \( k + 1 \) or more during \( C_2 \); that is, the counter size reaches \( z_1 + k + 1 \) at some point during \( C_2 \). Since \( M \) is normalized, there are at least \( k + 1 \) increasing steps and since \( M \) has \( k \) states, two must be in the same state. We can divide \( C_2 \):

\[
C_2 : (q_1, wb^n, z_1) \to^* (p, vxb^n, z) \to^* (p, xb^n, z + r) \to^* (q_2, b^n, z_2)
\]

where \( w = uvx \) and \( r > 0 \). Similarly, since \( z_2 \geq k + 1 \), the counter must drop by at least \( k + 1 \) during \( C_3 \), so there must be a segmentation

\[
C_3 : (q_2, b^n, z_2) \to^* (q, b^l, z') \to^* (q, b^{i-m}, z' - s) \to^* (f, e, 0),
\]

with \( s > 0 \). Furthermore, we can assume that during the first segment of \( C_3 \) the counter size is at least \( z' + 1 \) and during the second segment, at least \( z' - s + 1 \). Since \( M \) is normalized, \( m \geq s > 0 \).

Since the counter never becomes zero during \( C_2 \), we can pump it up without affecting the legitimacy of the computation. So, repeating \( us + 1 \) times, we have

\[
C_2' : (q_1, uw^s+1 xb^{n+mr}, z_1) \to^* (p, xb^{n+mr}, z + (s + 1) r) \to^* (q_2, b^{n+mr}, z_2 + rs).
\]

Similarly, the counter never becomes zero during the first two segments of \( C_3 \), so the same steps can be performed with a larger counter size. Thus, we have

\[
C_3' : (q_2, b^{n+mr}, z_2 + rs) \to^* (q, b^{i+mr}, z' + rs) \to^* (q, b^{i-m}, z' - s) \to^* (f, e, 0).
\]

Hence, putting together \( C_1, C_2' \) and \( C_3' \), we can obtain an accepting computation for \( a^n u w^s+1 x b^{n+mr} \), a contradiction, since \( mr \geq 1 \). \( \square \)

**Theorem 1:** The family of one counter languages is strongly resistant to chevron.

**Proof:** Let \( \langle L \rangle = L(M) \) for a one counter machine \( M \) with \( k \) states. Without loss of generality, we can assume that \( M \) is normalized. Let \( n \) be the integer given by lemma 2, \( k + 1 \leq n \leq k(k + 1) \). For any accepting computation for input \( a^n wb^n \), \( w \) in \( L \), the counter size does not exceed \( 2n + k \leq (2k + 1)(k + 1) + 1 \). One can construct from \( M \) a one counter machine \( M' \) which simulates all and only computations of \( M \) with counter size not exceeding \( (2k + 1)(k + 1) + 1 \). Clearly, \( L(M') \) is regular. Let \( T \) be the finite alphabet of \( L \) and \( L' = \{ w \text{ in } T^* | a^n wb^n \text{ is in } L(M') \} \). By definition of \( \langle L \rangle \), \( L' \subseteq L \). For any \( w \in L \), \( M' \) simulates an accepting computation for some word \( a^n wb^n \), so \( L = L' \). Hence, \( L \)
can be obtained from $L(M')$ by the homomorphism which erases $a$'s and $b$'s and is the identity elsewhere. Thus, $L$ is regular. On the other hand, if $L$ is regular, $\langle L \rangle$ is obviously a one counter language.

REFERENCES


