Theoretical Computer Science 23 (1983) 91–94
North-Holland Publishing Company

NOTE

ON SATISFYING THE LL-ITERATION THEOREM

Anton NIJHOLT

Department of Informatics, Faculty of Science, Katholieke Universiteit Nijmegen, The Netherlands

Communicated by M. Harrison
Received May 1982
Revised August 1982

Abstract. It is shown that the conditions in the iteration theorem for LL-languages are not sufficient to characterize these languages. Therefore the question raised by Beatty [1] can be answered negatively.

1. On satisfying the LL-iteration theorem

In Beatty [1] the question is raised whether satisfying the conditions of the iteration theorem for LL(k) languages is also sufficient to ensure that a language is LL(k). In this note it shown that this is not the case.

Beatty's iteration theorem for LL(k) languages is a special case of Ogden's Lemma for context-free languages (Ogden [8]). This note is in the tradition of several other notes in which it is shown that the conditions in an iteration theorem for a certain class of languages are not sufficient to characterize these languages (cf. Boasson [2], Boasson and Horvath [3], Horvath [4], Klove [5] and Nijholt [7]).

We start by recalling the LL-Iteration Theorem. The following notation is used. Let u be a string over an alphabet Σ. The length of u is denoted by $|u|$. Let k be a non-negative integer. For any u in Σ^*, $k:u$ denotes the prefix of u with length k if $|u|>k$ and u otherwise. In a similar way we define the notation $u:k$ for a suffix of u. Let L be an LL(k) language, where k is a non-negative integer. There exists an integer k_0 such that given a string w in L and k_0 or more distinguished positions (dp's) in w, we may write $w = w_1w_2w_3w_4w_5$ such that

(1) $w_2 \neq \varepsilon$.

(2) (a) Either w_1, w_2 and w_3 each contain dp's or w_3, w_4 and w_5 each contain dp's.

  (b) $w_2w_3w_4$ contains at most k_0 dp's.

(3) (a) Let $n = |w_1w_2|$ and suppose that w' is any string in L such that $n+k : w' = n+k : w$. Then there is a factorization $w_1w_2w_3'w_4'w_5'$ of w' such that

  (i) $w_1w_2^rw_3 \prod_{i=1}^{r} (u_i)w_5$,

(ii) $w_1 w_2^r w_3 \prod_{i=1}^{r} (u_i) w_5'$,

(iii) $w_1 w_2^r w_3' \prod_{i=1}^{r} (u_i) w_5$,

(iv) $w_1 w_2^r w_3' \prod_{i=1}^{r} (u_i) w_5'$

are in L for all $r \geq 0$ and for all strings $\prod_{i=1}^{r} (u_i)$ in which $u_i = w_4$ or $u_i = w_4'$, $1 \leq i \leq r$.

(b) Furthermore, if $\prod_{i=1}^{r} (\bar{u}_i)$ is a catenation of words $\bar{u}_i \in \{w_4, w_4'\}$ such that

$$\prod_{i=1}^{r} (u_i) = \prod_{i=1}^{r} (\bar{u}_i)$$

then $u_i = \bar{u}_i$, $1 \leq i \leq r$.

We want to show that there is a non-LL(k) language L and an integer k_0 such that for any $w \in L$ with k_0 or more dp's we can find a factorization $w_1 w_2 w_3 w_4 w_5$ of w such that the conditions (1), (2) and (3) are satisfied.

Firstly, we consider the special case with $k = 0$. Let $L = \{a, b\}^*$. Clearly, since any string over $\{a, b\}$ is in L, we can always find a factorization for $k_0 = 3$ such that the three conditions are satisfied. However, L is not an LL(0) language since it contains more than one element.

Next consider the case that $k > 0$. From Boasson and Horvath[3] we use the language

$$L = A_p \cup \Sigma^*\{aa, bb\}\Sigma^*$$

where $\Sigma = \{a, b\}$ and $A_P = \{(ab)^n \mid n \in P\}$ with P a subset of the natural numbers. P can always be chosen such that A_p, and therefore L, is not context-free. In [3] it has been observed that L is Ogden-like. That is, there exists an integer k_0 (in [3] $k_0 = 4$ has been used) such that given a string w in L and k_0 or more dp's in w we may write $w = w_1 w_2 w_3 w_4 w_5$ such that condition (2) is satisfied and, moreover, for each $r \geq 0$ the string $w_1 w_2^r w_3 w_4^r w_5$ is in L. We show that for $k_0 = 5$ we can always find a factorization such that the conditions (1), (2) and (3) are satisfied.

First we consider a string w in $\Sigma^*\{aa, bb\}\Sigma^*$. Hence, we can write $w = uaav$ (or the other case with $w = ubbv$, which can be treated similarly) for some u, v in Σ^*, ua not in $\Sigma^*\{aa, bb\}\Sigma^*$ and u not in $\Sigma^*\{aa, bb\}\Sigma^*$. Therefore, if $u \neq \varepsilon$, we can write $u = u'b$ with u' in Σ^*. We have the following case analysis. In each case the factorization $w_1 w_2 w_3 w_4 w_5$ of w is chosen in such a way that the conditions (1) and (2) are satisfied and such that $w_4 = \varepsilon$. For each of the chosen factorizations of w we will choose a factorization $w_1 w_2 w_3' w_4' w_5'$ of w' with $w_4' = w_5' = \varepsilon$. Therefore condition (3)(b) is always satisfied. Moreover, for each of the factorizations which will be mentioned it can easily be verified, using the property that $k : w_3 w_4 w_5 = k : w_3'$ ($k > 0$), that condition (3)(a) is satisfied.

Case 1: At most three dp's in uaa. For w_1 we take the longest prefix of w which contains three dp's. Hence, uaa is included in w_1; w_2 and w_3 can be chosen such that each contains exactly one dp, hence, $w_2 \neq \varepsilon$. Choose $w_4 = \varepsilon$ and let w_5 be the remainder of the string. Since aa is included in w_1 we have that $w_1 z$ is in $\Sigma^*\{aa, bb\}\Sigma^*$ for any string z in Σ^*. Therefore all the strings mentioned in (3)(a) are in L.

Case 2: More than three dp's in uaa.

Subcase 2.1: Two dp's in aa, at most three dp's in u. For this subcase we use the factorization $w_1 = = u'$, $w_2 = ba$, w_3 is the longest prefix of av which contains one dp, $w_4 = \varepsilon$ and w_5 is the remainder of the string. Notice that $w_1 : 1 = 1 : w_3$.

Subcase 2.2: One dp in aa, three dp's in u. It follows that $v \neq \varepsilon$.

2.2.1. If $1 : v = b$ then we take $w_1 = u$, $w_2 = aa$, w_3 is the longest prefix of v which contains one dp, $w_4 = \varepsilon$ and w_5 is the remainder of the string. Notice that $w_1 : 1 = 1 : w_3$.

2.2.2. If $1 : v = a$ then we take $w_1 = u'$, $w_2 = baa$, w_3 is the longest prefix of v which contains one dp, $w_4 = \varepsilon$ and w_5 is the remainder of the string. Notice that $w_1 : 1 = 1 : w_3$.

Subcase 2.3: More than three dp's in u.

2.3.1. If there are no dp's in v then we choose $w_4 = w_5 = \varepsilon$ and w_3 such that it includes aa and if aa has no dp's also exactly one of the dp's of u. Substring w_2 is chosen as the shortest string which has the property that it includes at least one dp and $1 : w_2 \neq 1 : w_3$. Notice that this is always possible. It follows that w_2 has at most two dp's and $1 : w_3 = w_1 : 1$. Substring w_1 is the remaining prefix of w.

2.3.2. If there are dp's in v, w_3 is longest prefix of v which contains one dp, $w_4 = \varepsilon$ and w_5 is the remainder of the string. We choose w_2 such that it includes aa, and in the case that aa has no dp's, a suffix of u which contains exactly one dp. It is always possible to choose w_2 such that $1 : w_2 \neq 1 : v$. Therefore $w_1 : 1 = 1 : w_3$.

It remains to show that also in the case that w is in A_p the three conditions are satisfied. Let w be in A_p. We can write $w = ubabv$ (or $w = uabav$, which can be treated in a similar way) with $w_1 = ub$, $w_2 = a$, where this a occurs at a distinguished position, and $w_3 = bv$, where bv is chosen in such a way that it contains exactly one dp. Hence, $w_4 = w_5 = \varepsilon$. Notice that $w_1 : 1 = 1 : w_3$. For any w' in L we use the factorization $w' = w_1 w_2 w_3'$. Hence, $w_4' = w_5' = \varepsilon$. Since $k : w_3' = k : w_3$ $(k > 0)$ it follows that $w_1 : 1 = 1 : w_3'$ and it immediately follows that we can satisfy the conditions (1), (2) and (3).

We may conclude that L satisfies the conditions of the LL-Iteration Theorem. Since L is not an LL(k) language these conditions are not sufficient for being LL.

Acknowledgment

I'm grateful to the referees for their comments on a first version of this paper.

References

[1] J.C. Beatty, Two iteration theorems for the LL(k) languages, *Theoret. Comput. Sci.* **12** (1980) 193–228.
[2] L. Boasson, A remark on Ogden's lemma, *EATCS-Bull.* **4** (1978) 3–4.
[3] L. Boasson and S. Horvath, On languages satisfying Ogden's lemma, *RAIRO Informat. Theor.* 12 (1978), 201–202.
[4] S. Horvath, The family of languages satisfying Bar-Hillel's lemma, *RAIRO Informat. Theor.* **12** (1978) 193–199.
[5] T. Klove, Pumping languages, *Internat J. Comput. Math.* **6** (1977) 115–125.
[6] A. Nijholt, An annotated bibliography of pumping, *EATCS-Bull.* **17** (1982) 34–53.
[7] A. Nijholt, A note on the sufficiency of Sokolowski's criterion for context-free languages, *Information Processing Lett.* **14** (1982) 207.
[8] W. Ogden, A helpful result for proving inherent ambiguity, *Math. Systems Theory* **2** (1968) 191–194.