Information Processing Letters 28 (1988) 33–37
North-Holland

30 May 1988

INFINITE HIERARCHY OF EXPRESSIONS CONTAINING SHUFFLE CLOSURE OPERATOR *

Joanna JĘDRZEJOWICZ

Institute of Mathematics, University of Gdańsk, ul. Wita Stwosza 57, 80-952 Gdańsk, Poland

Communicated by W.M. Turski
Received 21 September 1987
Revised 29 December 1987

Keywords: Shuffle operator, shuffle closure operator

1. Introduction

In [6] it was shown that languages generated by shuffle expressions with no nested ⊛ operator form a proper subclass of all the shuffle languages. Now we are going to show that the ⊛-depth hierarchy is infinite, that is, for each n there exists a language generated by an expression which contains exactly n nested ⊛ operators. The example of such a language is

$$L_n = (x_n L_{n-1} y_n z_n)^{\circledast},$$

where

$$L_1 = (x_1 y_1 z_1)^{\circledast}.$$

2. Preliminaries

Shuffle expressions SE and shuffle languages were investigated in [2,4,5,6], where the reader can find all necessary definitions.

Let Σ stand for an alphabet. We define the shuffle closure hierarchy (or, shortly ⊛-hierarchy) of shuffle expressions over Σ as follows.

2.1. Definition. SE^0 stands for the class of all regular expressions over the alphabet Σ.

For $n > 0$, SE^n is defined as follows:

(i) $\mathrm{SE}^{n-1} \subset \mathrm{SE}^n$,

(ii) if $P, Q \in \mathrm{SE}^n$, then $P \cdot Q$, $P + Q$, (P), $P \odot Q$, and P^* are all in SE^n,

(iii) if $P \in \mathrm{SE}^{n-1}$, then $P^{\circledast}$ is in SE^n,

(iv) nothing else is in SE^n.

Then $\mathrm{SE} = \bigcup_{n=0}^{\infty} \mathrm{SE}^n$, and as usual $L(\mathrm{SE}^n)$ stands for the class of languages generated by expressions from SE^n.

For each $n > 0$ we define a finite alphabet

$$\Sigma_1 = \{x_1, y_1, z_1\},$$

$$\Sigma_n = \Sigma_{n-1} \cup \{x_n, y_n, z_n\} \quad \text{for } n > 1.$$

Now, for a word $w \in \Sigma_n^*$ we define three predicates.

Let $\#_a w$ stand for the number of occurrences of a in w.

(P1) $\#_{x_i} w = \#_{y_i} w = \#_{z_i} w$ for $i = 1, 2, \ldots, n$,

(P2) for any prefix v of w,

$$\#_{x_i} v \geqslant \#_{y_i} v \geqslant \#_{z_i} v \quad \text{for } i = 1, 2, \ldots, n,$$

(P3) if $w = u x_k v$ for some k, satisfying $i \leqslant k < n$, then $\#_{x_i} u > \#_{y_i} u$ for each i, satisfying $k + 1 \leqslant i \leqslant n$.

We say that a shuffle expression R satisfies the predicates (P1), (P2), and (P3) of type n if $L(R) \subset \Sigma_n^*$ and each $w \in L(R)$ satisfies (P1), (P2), and

* This work was partially supported by Grant No. RP.I.09.

(P3). We omit 'of type n' if this is obvious from the context.

Let $h:\Sigma_n \to \Sigma_{n-1}$ be the homomorphism erasing the symbols x_n, y_n, and z_n; formally,

$$h(x)=\begin{cases}\varepsilon & \text{for } x\in\{x_n,\ y_n,\ z_n\},\\ x & \text{otherwise,}\end{cases}$$

where ε denotes the empty word.

We omit the proof of the following obvious lemma.

2.2. Lemma. *If every word $w \in L \subset \Sigma_n^*$ satisfies predicates* (P1), (P2), *and* (P3) *of type n, then every $w \in h(L) \subset \Sigma_{n-1}^*$ satisfies* (P1), (P2), *and* (P3) *of type $n-1$.*

As in [6], by a subexpression A of a shuffle expression R we mean any subword of R which itself is a shuffle expression.

For $w \in L(R)$, each subword of w generated by A is called a generator of w with respect to A. A formal definition is stated in [6].

Now we are going to prove a stronger form of [6, Lemma 2.1].

2.3. Lemma. *Let R satisfy* (P1) *and* (P2) *of type n. Then, for any subexpression $S^{\circledast}$ of R, every $\overline{w} \in L(S)$ satisfies* (P1) *and* (P2).

Proof. In [6, Lemma 2.1] it was proven that $\overline{w}$ satisfies (P1). Suppose that $\overline{w}$ does not satisfy (P2). Let $\overline{w}=vs$ and suppose that $\#_{x_i}v < \#_{y_i}v$ for some $1 \leqslant i \leqslant n$ (the proof for the case $\#_{y_i}v < \#_{z_i}v$ being similar).

As $\overline{w}=vs \in L(S)$ and $S^{\circledast}$ is a subexpression of R, there exist $q,\ r \in \Sigma_n^*$ such that

$$qv^ns^nr \in L(R) \quad \text{for every } n \in N.$$

Moreover, $\#_{x_i}qv \geqslant \#_{y_i}qv$, as $qvsr \in L(R)$ and thus satisfies (P2). Hence, $\#_{x_i}q > \#_{y_i}q$.

Since $qv^ns^nr \in L(R)$ for every $n \in N$, we have for big enough n (for example, $n > \#_{x_i}q - \#_{y_i}q$):

$$\#_{x_i}qv^n < \#_{y_i}qv^n,$$

a contradiction. Thus, qv^ns^nr satisfies (P2). □

3. Languages L_n

For every $n>0$ we define a language L_n over the alphabet Σ_n. We are going to show that $L_n \in L(\mathrm{SE}^n) - L(\mathrm{SE}^{n-1})$.

Let

$$L_1=(x_1y_1z_1)^{\circledast},$$

$$L_n=(x_nL_{n-1}y_nz_n)^{\circledast} \quad \text{for } n>1.$$

For every language L_n and for $k \in N$ we define an infinite family of special words Q_n^k:

$$Q_1^k=\{(x_1y_1)^rz_1^r:\ r>k\},$$

$$Q_n^k=\{x_nw_1y_n \ldots x_nw_ry_nz_n^r:\ w_1,\ldots,w_r \in Q_{n-1}^k,\ r>k\} \quad \text{for } n>1.$$

It is easily observed that $Q_n^k \subset L_n$ for $n \geqslant 1$. Other properties of L_n and Q_n^k are established in the following lemmas.

3.1. Lemma. *If $w \in L_n$, then w satisfies predicates* (P1), (P2), *and* (P3) *of type n.*

The easy, inductive proof of Lemma 3.1 is left to the reader.

3.2. Lemma. *Let R be an expression satisfying predicates* (P1), (P2), *and* (P3) *of type n and let $S^{\circledast}$ be a subexpression of R. If $(Q_n^k)^+ \cap L(R) \neq \emptyset$, then any prefix v of a generator of a word $w \in (Q_n^k)^+ \cap L(R)$ with respect to S satisfies*:

$$0 \leqslant \#_{x_i}v - \#_{y_i}v \leqslant 1 \quad \text{for } 1 \leqslant i \leqslant n.$$

Proof. $0 \leqslant \#_{x_i}v - \#_{y_i}v$ follows from Lemma 2.3.

Observe that for any prefix of a word from $(Q_n^k)^+$ the number of occurrences of x_i is either equal to that of y_i or 1 larger for any i, $1 \leqslant i \leqslant n$.

Now suppose that there exists a prefix v of a generator s of a word $w \in (Q_n^k)^+$ with respect to S satisfying

$$\#_{x_i}v - \#_{y}v > 1, \quad s=v\bar{s}.$$

The only way to generate w from s is to shuffle it with a word whose prefix contains at least one y_i more than x_i. Then, from the same generators, by shuffling in a different way, one obtains a word $\bar{w} \in L(R)$ which does not satisfy (P2). □

3.3. Lemma. *Let R be an expression satisfying predicates* (P1), (P2), *and* (P3) *of type n such that there exist $k \in N$ and $q \in (Q_n^k)^+ \cap L(R)$. Suppose $S^{\circledast}$ is a subexpression of R such that there exists a $\bar{w} \in L(S^{\circledast})$, which is a generator of q with respect to $S^{\circledast}$, and $\#_x \bar{w}_n > 0$. Then, every $w \in L(S)$ satisfies* (P3).

Proof. Let $\varepsilon \neq w \in L(S)$ and suppose w does not satisfy (P3). Thus, $w = ux_ky$ for certain k, $1 \leqslant k < n$, and $\#_{x_i} u = \#_{y_i} u$ for some i, $k+1 \leqslant i \leqslant n$ ($\#_{x_i} u \geqslant \#_{y_i} u$ follows from Lemma 2.3).

Since every word from $L(R)$ satisfies (P3), there exists an expression X such that $XS^{\circledast}$ is a subexpression of R and every word generated by X contains at least one occurrence of x_i more than that of y_i.

Thus there exists a word $x \in L(X)$ such that:

(i) $x\bar{w}$ is a generator of q with respect to $XS^{\circledast}$,

(ii) $\#_{x_i} x - 1 \geqslant \#_{y_i} x$.

In addition,

(iii) $\#_{x_n} \bar{w} = \#_{y_n} \bar{w} = \#_{z_n} \bar{w} > 0$ follows from Lemma 2.3 and the assumptions of this lemma, and

(iv) $\#_{x_i} \bar{w} = \#_{y_i} \bar{w}$ for all $1 \leqslant i \leqslant n$ follows from Lemma 2.3.

Observe that, for any word $v \in (Q_n^k)^{\circledast}$ containing z_n, the prefix of v preceding z_n has the same number of x_i's and y_i's for each $1 \leqslant i \leqslant n$.

From (ii) and (iv) it follows that the only possibility to generate q from xs is to shuffle it with a word p whose prefix contains at least one y_i more than x_i. This is a contradiction as the word beginning with p belongs to the same shuffle and, thus, belongs to $L(R)$ but does not satisfy (P2). □

Now, using these lemmas we are going to prove our main result.

3.4. Theorem. *For every $n > 0$, if $R \in \mathrm{SE}^{n-1}$, $L(R) \subset \Sigma_n^*$ and R satisfies predicates* (P1), (P2), *and* (P3) *of type n, then there exists a $k \in N$ such that*

$$L(R) \cap (Q_n^k)^+ = \emptyset.$$

Proof. The proof is by induction on n and uses the ideas of [6].

Basis: For $n = 1$ we define $k = nstates + 1$, where *nstates* is the number of states of a finite automaton accepting the regular language $L(R)$.

Induction step: Suppose that if $S \in \mathrm{SE}^{n-2}$, $L(S) \subset \Sigma_{n-1}^*$ and S satisfies (P1), (P2), and (P3) of type $n-1$, then there exists a $k \in N$ such that

$$L(S) \cap (Q_{n-1}^k)^+ = \emptyset.$$

Now let $R \in \mathrm{SE}^{n-1}$, $L(R) \subset \Sigma_n^*$ and suppose that R satisfies (P1), (P2), and (P3) of type n.

Let $S_1, \ldots, S_p$ be all the subexpressions of R from SE^{n-2} such that words from the languages generated by $h(S_1), \ldots, h(S_p)$ satisfy (P1), (P2), and (P3) of type $n-1$ (h is the homomorphism erasing x_n, y_n, and z_n—see Lemma 2.2). From the induction hypothesis it follows that for each S_i, $1 \leqslant i \leqslant p$, there exists a constant k_i such that

$$L(h(S_i)) \cap (Q_{n-1}^{k_i})^+ = \emptyset.$$

Let $k = \max\{k_1, \ldots, k_p, |R| + 1\}$ and suppose that there exists a $w \in L(R) \cap (Q_n^k)^+$.

Let $S^{\circledast}$ be a subexpression of R and let s be a generator of w with respect to $S^{\circledast}$. Thus,

$$s \in L(s_1 \odot \cdots \odot s_m) \quad \text{for } s_i \neq \varepsilon,\ s_i \in L(S),\ 1 \leqslant i \leqslant m.$$

Using Lemma 3.3 it can easily be established that either $\#_{x_n} s_i = 0$ for $i = 1, 2, \ldots, m$ or $\#_{x_n} s_i > 0$, for $i = 1, 2, \ldots, m$.

In the former case, we call $S^{\circledast}$ to be of type 0 and, in the latter, to be of type 1. Thus, either:

(1) all subexpressions $S^{\circledast}$ of R are of type 0, or

(2) there exists a subexpression $S^{\circledast}$ of type 1.

Now we are going to examine each case in detail and find a contradiction in each of them.

(1) Since $w \in (Q_n^k)^+$, we have $w = \bar{w} \cdot \bar{\bar{w}}$ for some $\bar{w} \in Q_n^k$ and $\bar{\bar{w}} \in (Q_n^k)^*$. Hence,

$$\bar{w} = x_n w_1 y_n \ldots x_n w_r y_n z_n^r \quad \text{for some } r > k$$

and

$$w_1, \ldots, w_r \in Q_{n-1}^k.$$

Since $r > |R|$ and x_n appears in R at most $|R|$

times, at least two occurrences of x_n in $\bar{w}$ are generated by some expression of type S^*.

Strictly speaking there exist a subexpression of S^* of R and a generator s of w with respect to S^* such that

$$s = s_1 \dots s_k, \quad k \geqslant 2,$$

and there exist $i < j$ such that

$$\#_{x_n} s_i > 0 \quad \text{and} \quad \#_{x_n} s_j > 0.$$

From [6, Lemma 2.1] it follows that

$$\#_{x_n} s_i = \#_{z_n} s_i > 0.$$

Thus, z_n precedes s_j (and x_n) in $\bar{w}$ which is a contradiction.

(2) Let S be a subexpression of R of type 1 and v a generator of w with respect to S. We shall show that $h(v) \in (Q_{n-1}^k)^+$.

From Lemmas 2.3, 3.2, and 3.3 it follows that

$$v = x_n s_1 y_n t_1 \dots x_n s_p y_n t_p \quad \text{for some } p > 1,$$

and $s_1, \dots, s_p$, $t_1, \dots, t_p$ do not contain x_n and y_n.

Observe that

(i) $s_j \in Q_{n-1}^k$ for $j = 1, 2, \dots, p$.

The symbol x_n preceding s_j in v appears in w and precedes some word $w_{m1} \in Q_{n-1}^k$ in w. Hence,

$$w = A \ x_n w_{m1} y_n B$$

and $\updownarrow$

$$v = \dots x_n s_j y_n \quad \dots .$$

Suppose $s_j \neq w_{m1}$. Then, w is generated from v and some word $\bar{v}$ containing a nonempty $b \in \Sigma_{n-1}^*$, using the shuffle operator as indicated by the following figure:

$$v = \dots x_n s_j y_n \dots \quad ; \quad \bar{v} = \dots b \dots$$
$$w = A x_n w_{m1} y_n B.$$

The same shuffle (thus the language $L(R)$ as well) contains a word

$$\bar{w} = A b x_n \bar{w}_{m1} y_n B,$$

where $\bar{w}_{m1}$ is formed from w_{m1} by erasing all the letters of b.

But, if $\bar{w} \in L(R)$, then $\bar{w}$ satisfies (P2) and (P3) and this is not the case since

$$\#_{x_i} A = \#_{y_i} A = \#_{z_i} A, \quad i = 1, 2, \dots, n,$$
$$\#_{x_n} b = 0, \quad b \neq \varepsilon.$$

Now we show that

(ii) $h(t_j) = \varepsilon$ for $j = 1, 2, \dots, p$.

As we have shown in (i), $s_1, \dots, s_p \in Q_{n-1}^k$ and, from Lemmas 2.3 and 3.3 it follows that v satisfies (P1), (P2), and (P3). Thus, $t_1, \dots, t_p$ do not contain $x_{n-1}, x_{n-2}, \dots, x_1$ and therefore $t_1, \dots, t_p$ do not contain $y_{n-1}, \dots, y_1$, $z_{n-1}, \dots, z_1$ as well. Hence (ii).

From (i) and (ii) we have

(iii) $h(v) \in (Q_{n-1}^k)^+$.

From Lemmas 2.3 and 3.3 it follows that each $w \in L(S)$ satisfies (P1), (P2), and (P3) of type n. Hence, from Lemma 2.2, words from the language generated by $h(S)$ satisfy (P1), (P2), and (P3) of type $n-1$. Moreover, if $R \in \mathrm{SE}^{n-1}$, then $S \in \mathrm{SE}^{n-2}$.

Now, using the inductive assumption we obtain

$$L(h(S)) \cap (Q_{n-1}^k)^+ = \emptyset$$

(since $k_1, \dots, k_p > k$) and this is a contradiction to (iii). □

3.5. Theorem

$L_n \in L(\mathrm{SE}^n) - L(\mathrm{SE}^{n-1})$ *for* $n > 0$.

Proof. The proof immediately follows from Lemma 3.1 and Theorem 3.4. □

3.6. Remark. We have thus established that for every n there exists a language L_n over a finite alphabet Σ_n such that each expression generating L_n contains at least n nested ⊛ operators. This demonstrates that the ⊛-hierarchy over an infinite alphabet is infinite. It is an open problem whether there exists a finite alphabet containing more than one letter over which the hierarchy is infinite.

Acknowledgment

The author would like to thank two anonymous referees for their careful reading of the manuscript and for giving several suggestions that improved the presentation of the paper.

References

[1] T. Araki, T. Kagimasa and N. Tokura, Relations of flow languages to Petri net languages, *Theoret. Comput. Sci.* **15** (1) (1981) 51–75.

[2] J. Gischer, Shuffle languages, Petri nets and context-sensitive grammars, *Comm. ACM* **24** (1981) 597–605.

[3] M. Hack, Petri net languages, *Computation Structures Group Memo 124*, Project MAC, MIT, 1975.

[4] M. Jantzen, Extending regular expressions with iterated shuffle, *Theoret. Comput. Sci.* **38** (1985) 223–247.

[5] J. Jędrzejowicz, On the enlargement of the class of regular languages by shuffle closure, *Inform. Process. Lett.* **16** (1983) 51–54.

[6] J. Jędrzejowicz, Nesting of shuffle closure is important, *Inform. Process. Lett.* **25** (1987) 363–367.