On the Decidability of 2-Infix-Outfix Codes

School of Computing Technical Report 2004-479

Michael Domaratzki[*]
School of Computing
Queen's University, Kingston, ON K7L 3S6
email: domaratz@cs.queensu.ca

April 6, 2004

1 Introduction

The theory of codes is a fertile area at the intersection of formal language theory, error detection and correction, data compression and data security [6]. Theoretical research into codes is often interested with combinatorial properties of formal languages related to codes.

In particular, there has been substantial recent interest in classes of codes defined by certain "finite subset" conditions. In general, given a class $\mathcal{C}$ of codes and $m \geq 0$, we may define a class $\mathcal{C}'_m$ as follows:

$$L \in \mathcal{C}'_m \iff (L' \subseteq L, |L'| \leq m \Rightarrow L' \in \mathcal{C}).$$

Thus, for instance, a language L is an n-*code* if every language $L' \subseteq L$ of size at most n is a code [5]. Also studied are n-prefix-suffix codes [3], n-infix-outfix codes [8, 9, 7], n-intercodes [6, p. 555] and others. A general framework for defining such "finite subset" classes of languages is given, e.g., by Jürgensen and Konstantinidis [6, pp. 565–567].

Decidability problems for such classes of languages appear to be very difficult. It is an open problem whether a regular language is an n-code for $n > 2$ [6,

[*] Research supported in part by an NSERC PGS-B graduate scholarship.

Table 9.1]. This problem is one of the most easily stated open problems in all of formal language theory, and is of fundamental interest to the entire field as well.

In this note, we investigate the decidability of 2-infix-outfix codes, introduced by Long *et al.* [8, 9, 7]. We first show that it is decidable whether a regular language is a 2-infix-outfix code. This result is an extension of a result on 2-prefix-suffix codes due to Ito *et al.* [3]. To complement the positive decidability result, we also show that it is undecidable whether an arbitrary linear context-free language (LCFL) is a 2-infix-outfix code.

2 Preliminary Definitions

For a background on regular languages and formal language theory, please see Yu [10]. Let Σ be a finite set of symbols, called *letters*. Then Σ^* is the set of all finite sequences of letters from Σ, which are called *words*. The empty word ϵ is the empty sequence of letters. The *length* of a word $w = w_1 w_2 \cdots w_n \in \Sigma^*$, where $w_i \in \Sigma$, is n, and is denoted $|w|$. Note that ϵ is the unique word of length 0. Given a word $w \in \Sigma^*$ and $a \in \Sigma$, $|w|_a$ is the number of occurrences of a in w. A *language* L is any subset of Σ^*.

A *deterministic finite automaton* (DFA) is a five-tuple $M = (Q, \Sigma, \delta, q_0, F)$ where Q is a finite set of states, Σ is an alphabet, $\delta : Q \times \Sigma \to Q$ is a transition function, $q_0 \in Q$ is the start state, and $F \subseteq Q$ is the set of final states. We extend δ to $Q \times \Sigma^*$ in the usual way. A word $w \in \Sigma^*$ is accepted by M if $\delta(q_0, w) \in F$. The *language accepted* by M, denoted $L(M)$, is the set of all words accepted by M. A language is called *regular* if it is accepted by some DFA. Given a regular language L, the *state complexity* of L, denoted $\mathrm{sc}(L)$, is the minimal number of states in any DFA accepting L.

Let $\leq_d$ be the binary relation on Σ^* defined by $u \leq_d v$ iff there exist $x, y \in \Sigma^*$ such that $ux = yu = v$. Let ω_{io} be the binary relation defined on Σ^* by $u \ \omega_{io} \ v$ iff there exist a factorization $u = u_1 u_2$ and words y_1, y_2, x such that $v = u_1 x u_2 = y_1 u y_2$. Note that $u \ \omega_{io} \ v$ iff $u \leq_i v$ and $u \ \omega_o \ v$, where $\leq_i$ and ω_o are the infix ordering and the outfix relation, respectively; see Ito *et al.* [4].

Recall that a set S is an antichain under a binary relation ω if $x \omega y$ and $x, y \in S$ implies $x = y$. The following characterization relates the binary relations $\leq_d, \omega_{io}$ with the classes of 2-prefix-suffix and 2-infix-outfix codes:

Lemma 2.1 *Let L be a language. Then L is an anti-chain under $\leq_d$ (resp., ω_{io}) iff L is a 2-prefix-suffix code (resp., a 2-infix-outfix code).*

For more information on prefix, suffix, infix and outfix codes, as well as n-prefix-suffix code, see Jürgensen and Konstantinidis [6]. The class of n-infix-outfix codes was introduced by Long and others (see, e.g., Long [7], Long *et al.* [8, 9]). Our main result is that given a regular language L, it is decidable whether L is a 2-infix-outfix code.

3 Decidability

We will require some preliminary results. The following is a restatement of a result due to Ito *et al.* [3, Lemma 7.2]:

Lemma 3.1 *Let L be a regular language and let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA accepting L with $|Q| = n$. Let $u, v \in \Sigma^*$ be words satisfying $u \leq_d v$. Then there exist $u', v' \in \Sigma^*$ such that the following conditions hold:*

(a) $u' \leq_d v'$;

(b) for all $q \in Q$, $\delta(q, u) = \delta(q, u')$ (resp., $\delta(q, v) = \delta(q, v')$);

(c) $|v'| \leq n^{n+3} + 3n^{n+2} + n^{n+1} - n - 2$.

Further, if $u \neq v$ then $u' \neq v'$.

It is not known if the bound given in (c) is optimal. Let $f : \mathbb{N} \to \mathbb{N}$ be the function $f(n) = n^{n+3} + 3n^{n+2} + n^{n+1} - n - 2$.

We now give an interesting relation between $\leq_d$ and ω_{io}:

Lemma 3.2 *Let $u, v \in \Sigma^*$. Then $u \, \omega_{io} \, v$ iff there exist factorizations $u = u_1u_2$ and $v = v_1v_2$ such that $u_i \leq_d v_i$ for $i = 1, 2$.*

Proof. Let $u \, \omega_{io} \, v$. Then $v = u_1xu_2 = y_1uy_2$. Note that $y_1uy_2 = y_1u_1u_2y_2$. As $|x| = |y_1| + |y_2|$, let $x = x_1x_2$ where $|x_i| = |y_i|$ for $i = 1, 2$. Note that $v = u_1x_1x_2u_2 = y_1u_1u_2y_2$. Thus, $u_1x_1 = y_1u_1$ and $x_2u_2 = u_2y_2$. Therefore, let $v_1 = u_1x_1$ and $v_2 = u_2y_2$. The implication follows.

For the reverse implication, assume that $u = u_1u_2$ and $v = v_1v_2$ such that $u_i \leq_d v_i$ for $i = 1, 2$. Let x_i, y_i for $i = 1, 2$ be such that $u_ix_i = y_iu_i = v_i$. Then $v = v_1v_2 = u_1x_1y_2u_2 = y_1u_1u_2x_2$. Thus, $u \, \omega_{io} \, v$. ■

We now extend the characterization of Lemma 3.1 to ω_{io}:

Lemma 3.3 *Let L be a regular language and let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA accepting L with $|Q| = n$. Let $u, v \in \Sigma^*$ be words satisfying $u\, \omega_{io}\, v$. Then there exists $u', v' \in \Sigma^*$ such that the following conditions hold:*

(a) $u'\, \omega_{io}\, v'$*;*

(b) for all $q \in Q$, $\delta(q, u) = \delta(q, u')$ (resp., $\delta(q, v) = \delta(q, v')$);

(c) $|v'| \leq 2f(n)$.

Further, if $u \neq v$ then $u' \neq v'$.

Proof. Let $u, v \in L$ with $u \neq v$ and $u\, \omega_{io}\, v$. Let $u = u_1u_2$ and $v = v_1v_2$ such that $u_i \leq_d v_i$ for $i = 1, 2$.

Let $q_u, q_v \in Q$ be arbitrary. Let $q_1, q_2, r_1, r_2 \in Q$ be chosen so that $\delta(q_u, u_1) = q_1$, $\delta(q_1, u_2) = q_2$, $\delta(q_v, v_1) = r_1$, and $\delta(r_1, v_2) = r_2$. Note that $\delta(q_u, u) = q_2$ and $\delta(q_v, v) = r_2$.

Consider u_1, v_1. As $u_1 \leq_d v_1$, by Lemma 3.1 there exist u_1', v_1' such that $u_1' \leq_d v_1'$, $\delta(q_u, u_1') = q_1$, $\delta(q_v, v_1') = r_1$, and $|v_1'| \leq f(n)$.

Consider now u_2, v_2. As $u_2 \leq_d v_2$, there exist u_2', v_2' such that $u_2' \leq_d v_2'$, $\delta(q_1, u_2') = q_2$, $\delta(r_1, v_2') = r_2$, and $|v_2'| \leq f(n)$, again by Lemma 3.1.

Let $u' = u_1'u_2'$ and $v' = v_1'v_2'$. Note $\delta(q_u, u') = q_2$ and $\delta(q_v, v') = r_2$. Further, by Lemma 3.2, $u'\, \omega_{io}\, v'$. Thus, (a) and (b) hold. Condition (c) holds as $|v'| = |v_1'| + |v_2'| \leq 2f(n)$.

If $u \neq v$, then one of $u_1 \neq v_1$ or $u_2 \neq v_2$ holds, and thus $u_1' \neq v_1'$ or $u_2' \neq v_2'$ holds. In particular, $u' \neq v'$. ■

Corollary 3.4 *Let L be a regular language with $\mathrm{sc}(L) = n$. If there exist distinct $u, v \in L$ such that $u\, \omega_{io}\, v$, then there exist distinct $u', v' \in L$ with $u'\, \omega_{io}\, v'$ and $|v'| \leq 2f(n)$.*

Proof. Let $L = L(M)$ with $M = (Q, \Sigma, \delta, q_0, F)$ and $|Q| = n$. Let $u, v \in L$ be distinct words such that $u\, \omega_{io}\, v$. Let $\delta(q_0, u) = q_1$ and $\delta(q_0, v) = q_2$. Note that $q_1, q_2 \in F$. By Lemma 3.3, there exist distinct u', v' such that $u'\, \omega_{io}\, v'$ and $|v'| \leq 2f(n)$. Further, $\delta(q_0, u') = q_1 \in F$ and $\delta(q_0, v') = q_2 \in F$. This establishes the corollary. ■

Our main theorem is now immediate:

Theorem 3.5 *Let L be a regular language. Then it is decidable whether L is a 2-infix-outfix code.*

Proof. It suffices to check all distinct $u, v \in L$ with $|u| < |v| \leq 2n$ for $u \, \omega_{io} \, v$. ■

4 Undecidability

We complement the decidability result of the previous section with the following undecidability result. Our reduction is from Post's Correspondence Problem (PCP); an introduction to PCP is given by Harju and Karhumäki [2]. For the formal definitions of LCFLs, see Autebert *et al.* [1].

Theorem 4.1 *Given an LCFL L, it is undecidable whether L is a 2-infix-outfix code.*

Proof. Let $P = (u_1, \ldots, u_n; v_1, \ldots, v_n)$ be a PCP instance over Σ. Let $0, 1, \$ \notin \Sigma$ and define the languages $L_1, L_2 \subseteq (\Sigma \cup \{0, 1, \$\})^*$ as follows:

$$\begin{aligned} L_1 &= \{\$u_{i_1} \cdots u_{i_m}\$\$0^{i_m}1 \cdots 0^{i_1}1\$ \, : \, m \geq 1, 1 \leq i_p \leq n, 1 \leq p \leq m\}, \\ L_2 &= \{\$v_{j_1} \cdots v_{j_r}\$u_{i_1} \cdots u_{i_m}\$\$0^{i_m}1 \cdots 0^{i_1}1\$0^{j_r}1 \cdots 0^{j_1}1\$ \\ &\qquad : \, m, r \geq 1, 1 \leq i_p, j_s \leq n, 1 \leq p \leq m, 1 \leq s \leq r\}. \end{aligned}$$

Let $L = L_1 \cup L_2$. Note that L_1, L_2 and L are LCFLs. We claim that L is a 2-infix-outfix code iff P has no solutions. This will establish the result.

Assume that P has a solution. Let $m \geq 1$ and $1 \leq i_j \leq n$ for $1 \leq j \leq m$ be such that $u_{i_1} \cdots u_{i_m} = v_{i_1} \cdots v_{i_m}$. Let $\alpha = u_{i_1} \cdots u_{i_m} = v_{i_1} \cdots v_{i_m}$. Let $\beta = 0^{i_m}1 \cdots 0^{i_1}1$. Then note that $x = \$\alpha\$\$\beta\$$ and $y = \$\alpha x \beta\$$ satisfy $x, y \in L$ and $x \neq y$. Further, $x \, \omega_{io} \, y$. Thus, L is not a 2-infix-outfix code.

Now assume that L is not a 2-infix-outfix code. Then there exist $x, y \in L$ such that $x \neq y$ and $x \, \omega_{io} \, y$. There are four cases:

(a) $x, y \in L_1$. Then $x = \$\alpha_1\$\$\alpha_2\$$ and $y = \$\beta_1\$\$\beta_2\$$ for some $\alpha_i, \beta_i \in (\Sigma \cup \{0, 1\})^*$, $i = 1, 2$. Thus, we must have that $x = y$, a contradiction.

(b) $x, y \in L_2$. In this case, there exist $\alpha_i, \beta_i \in (\Sigma \cup \{0, 1\})^*$ with $1 \leq i \leq 4$ such that $x = \$\alpha_1\$\alpha_2\$\$\alpha_3\$\alpha_4\$$ and $y = \$\beta_1\$\beta_2\$\$\beta_3\$\beta_4\$$. Again, we have that $x = y$.

(c) $x \in L_2$ and $y \in L_1$. In this case, $|x|_\$ = 6$ and $|y|_\$ = 4$, which is impossible if $x\ \omega_{io}\ y$.

(d) $x \in L_1$ and $y \in L_2$. Then we can write

$$\begin{aligned} x &= \$u_{i_1}u_{i_2}\cdots u_{i_m}\$\$0^{i_m}1\cdots 0^{i_1}1\$, \\ y &= \$v_{j_1}\cdots v_{j_s}\$u_{k_1}u_{k_2}\cdots u_{k_\ell}\$\$0^{k_\ell}1\cdots 0^{k_1}\$0^{j_s}1\cdots 0^{j_1}1\$, \end{aligned}$$

where $m, s, \ell \geq 1$, $1 \leq i_p, j_q, k_r \leq n$ for $1 \leq p \leq m$, $1 \leq q \leq s$ and $1 \leq r \leq \ell$. As $x\ \omega_{io}\ y$, there exist α_1, α_2 such that $y = \alpha_1 x \alpha_2$. Therefore, we must have that the occurrences of the subword \$\$ match between x and y, that $\alpha_1 = \$v_{j_1}\cdots v_{j_2}$, $\alpha_2 = 0^{j_s}1\cdots 0^{j_1}1\$$ and, further, $m = \ell$ and $i_p = k_p$ for $1 \leq p \leq m$. Thus,

$$y = \$v_{j_1}\cdots v_{j_s}\$u_{i_1}\cdots u_{i_m}\$\$0^{i_m}1\cdots 0^{i_1}1\$0^{j_s}1\cdots 0^{j_1}1\$.$$

Now, there also exist x_1, x_2, β such that $x = x_1 x_2$ and $y = x_1 \beta x_2$. As $\$, 0, 1 \notin \Sigma$, we must have that

$$\begin{aligned} x_1 &= \$v_{j_1}\cdots v_{j_s}\$, \\ x_2 &= \$0^{j_s}1\cdots 0^{j_1}1\$, \\ \$\beta\$ &= x. \end{aligned}$$

We must necessarily have that $x_1 = \$u_{i_1}\cdots u_{i_m}\$$ and $x_2 = \$0^{i_m}\cdots 0^{i_1}\$$. Thus, $s = m$, $j_q = i_q$ for all $1 \leq q \leq s$, and

$$v_{j_1}\cdots v_{j_s} = u_{j_1}\cdots u_{j_s}.$$

Therefore, P has a solution.

This establishes the result. ■

We note that a similar construction can establish the undecidability of determining whether an LCFL is a 2-prefix-suffix code, which was apparently not considered by Ito *et al.* [3].

5 Conclusion

In this note, we have considered decidability problems related to 2-infix-outfix codes. The problem for more general classes, such as n-k-infix-outfix codes and

n-k-prefix-suffix codes introduced by Long *et al.* [8, 7], as well as, more crucially, n-codes, appear to still be open.

Further, the proof techniques used here are yet another example of ad-hoc methods for proving decidability. Unfortunately, the general methods discussed in Jürgensen and Konstantinidis [6, Sect. 9] do not appear to be applicable to these situations. It remains a substantial challenge to find general classes of languages defined by such conditions to which uniform decidability results apply.

References

[1] AUTEBERT, J.-M., BERSTEL, J., AND BOASSON, L. Context-free languages and pushdown automata. In *Handbook of Formal Languages, Vol. I,* G. Rozenberg and A. Salomaa, Eds. Springer-Verlag, 1997, pp. 111–174.

[2] HARJU, T., AND KARHUMÄKI, J. Morphisms. In *Handbook of Formal Languages, Vol. I* (1997), G. Rozenberg and A. Salomaa, Eds., Springer-Verlag, pp. 439–510.

[3] ITO, M., JÜRGENSEN, H., SHYR, H., AND THIERRIN, G. n-prefix-suffix languages. *Intl. J. Comp. Math. 30* (1989), 37–56.

[4] ITO, M., JÜRGENSEN, H., SHYR, H., AND THIERRIN, G. Outfix and infix codes and related classes of languages. *J. Comp. Sys. Sci. 43* (1991), 484–508.

[5] ITO, M., JÜRGENSEN, H., SHYR, H., AND THIERRIN, G. Languages whose n-element subsets are codes. *Theor. Comput. Sci. 96* (1992), 325–344.

[6] JÜRGENSEN, H., AND KONSTANTINIDIS, S. Codes. In *Handbook of Formal Languages, Vol. I,* G. Rozenberg and A. Salomaa, Eds. Springer-Verlag, 1997, pp. 511–600.

[7] LONG, D. *Study of Coding Theory and its Application to Cryptography*. PhD thesis, City University of Hong Kong, 2002.

[8] LONG, D., JIA, W., MA, J., AND ZHOU, D. k-p-infix codes and semaphore codes. *Disc. Appl. Math. 109* (2001), 237–252.

[9] LONG, D., MA, J., AND ZHOU, D. Structure of 3-infix-outfix maximal codes. *Theor. Comp. Sci. 188* (1997), 231–240.

[10] YU, S. Regular languages. In *Handbook of Formal Languages, Vol. I*, G. Rozenberg and A. Salomaa, Eds. Springer-Verlag, 1997, pp. 41–110.