Watson-Crick Conjugates of Words and Languages

Kalpana Mahalingam [1]   Anuran Maity[1,2,3*]

[1] *Department of Mathematics, Indian Institute of Technology Madras, Chennai, India*
[2] *Saint Petersburg State University, 7/9 Universitetskaya nab., St. Petersburg, Russia*
[3] *Department of Mathematics, Indian Institute of Technology Guwahati, Guwahati, India*

In this work, we explore the concept of Watson-Crick conjugates, also known as θ-conjugates (where θ is an antimorphic involution), of words and languages. This concept extends the classical idea of conjugates by incorporating the Watson-Crick complementarity of DNA sequences. Our investigation initially focuses on the properties of θ-conjugates of words. We then define θ-conjugates of a language and study closure properties of certain families of languages under the θ-conjugate operation. Furthermore, we analyze the iterated θ-conjugate of both words and languages. Finally, we discuss the idea of θ-conjugate-free languages and examine some decidability problems related to it.

1   Introduction

Conjugates of a word are significant in the field of combinatorics on words as they have numerous applications in cryptography (Belkheyar et al. (2023)), biology (Walter et al. (2015)), and other areas. Our analysis in this work focuses on various combinatorial, structural, and language-theoretic properties of Watson-Crick conjugates, or θ-conjugates (where θ is an antimorphic involution) of words - a generalization of the conjugates of words.

Over the years, DNA sequences have been used to encode/design a range of complex problems/experiments (for example, see Adleman (1994), Aoi et al. (1998), Bi et al. (2016), Braich et al. (2002), Head et al. (2000), Henkel et al. (2007), Henkel et al. (2005), Jeddi and Saiz (2017), Johnson (2008), Ouyang et al. (1997), Sakamoto et al. (2000), Shi et al. (2017), Wang et al. (2008), Zhang and Seelig (2011), etc). When encoding or designing a problem using DNA sequences, the occurrence of inter/intra-molecular hybridizations (see Fig. 1) between DNA strands can be perceived both as an advantage and a disadvantage. For example, formation of hairpin structures have found utility in various applications, including solving the satisfiability problem (Sakamoto et al. (2000)), designing DNA aptamers (Jeddi and Saiz (2017)),

---

*Corresponding author

arXiv:2208.03123v5 [cs.FL] 27 Aug 2025

and programming catalytic DNA self-assembly strategies (Bi et al. (2016)). Conversely, these hybridiza-
tions are sometimes viewed as a disadvantage (see Shi et al. (2017); Zhang and Seelig (2011)). This
is primarily because these hybridizations can significantly disrupt the intended interactions between the
involved sequences in inter/intra-molecular hybridizations and other DNA strands in the preprogrammed
and expected ways. Therefore, to avoid such undesirable hybridizations, careful DNA sequence design is
necessary at the time of encoding the problem. Drawing inspiration from this approach of DNA sequence
design, several authors have proposed various conditions for encoding by extending various concepts
of classical combinatorics on words, such as primitive words (Chiniforooshan et al. (2010), Czeizler
et al. (2010), Gawrychowski et al. (2014), Kari and Kulkarni (2014a), Kari and Ng (2019)), bordered
words (Kari and Kulkarni (2014b), Kari and Kulkarni (2017), Kari and Mahalingam (2007), Kari and
Mahalingam (2008a), Kari and Mahalingam (2008b), Kari and Seki (2009)), palindromes (Kari and Ma-
halingam (2008b), Kari and Mahalingam (2010)), and rich words (Starosta (2011)).
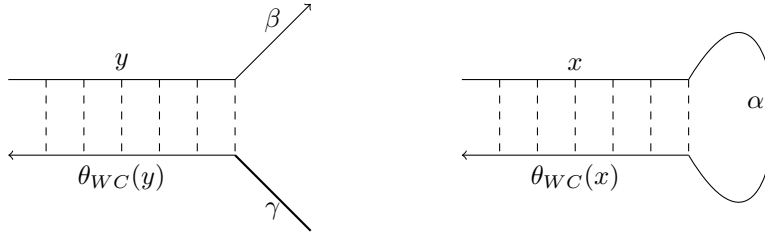

Fig. 1: Example of inter- and intra- molecular structures

In Kari and Mahalingam (2007), the authors extended the concept of bordered word to θ-bordered word
from the perspective of DNA computing, where θ is an antimorphic involution (i.e., $\theta(xy) = \theta(y)\theta(x)$
and $\theta^2(t) = t$ for any $x, y, t \in \Sigma^*$) that incorporates the notion of the Watson-Crick complementarity of
DNA molecules. They also defined θ-bordered word for a morphic involution θ (i.e., $\theta(xy) = \theta(x)\theta(y)$
and $\theta^2(t) = t$ for any $x, y, t \in \Sigma^*$) as a generalization of bordered word. For a morphic/antimorphic
involution θ, a word w is said to be θ-bordered if $w = uz = \theta(z)v$ for some non-empty words u, z, v.
A word that is not θ-bordered is called θ-unbordered. When focusing on the DNA alphabet $\{A, T, G, C\}$
and utilizing the Watson-Crick complementarity function θ_{WC} (θ_{WC} is an antimorphic involution on
$\{A, T, G, C\}^*$ such that $\theta_{WC}(A) = T$, $\theta_{WC}(T) = A$, $\theta_{WC}(C) = G$, $\theta_{WC}(G) = C$), θ_{WC}-unbordered
words represents the collection of DNA single strands that are free from both inter- and intra-molecular
structures such as the ones shown in Fig. 1(left), and hairpins shown in Fig. 1(right). From the defini-
tion of θ-bordered word, Kari and Mahalingam (2008b) defined the notion of θ-conjugate of a word for
morphic/antimorphic involution θ. Formally, a word u is called a θ-conjugate of a word v if there exists
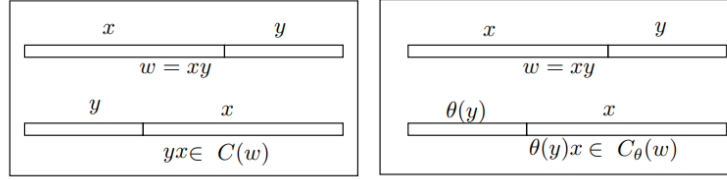a word z such that $uz = \theta(z)v$. The set of all θ-conjugates of v is denoted by $C_\theta(v)$. When θ is an

Fig. 2: Structure of conjugates & θ-conjugates of word for antimorphic involution θ

antimorphic involution over Σ^*, then analyzing the solution of the equation $uz = \theta(z)v$ (as presented in Proposition 2 in Kari and Mahalingam (2008b)), it becomes evident that for $v = xy$, the θ-conjugate u can be expressed as $\theta(y)x$ for some $x, y \in \Sigma^*$ (see Fig. 2). In this scenario, if we consider the DNA alphabet
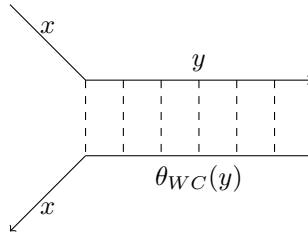

Fig. 3: Inter-molecular structure between a word and its θ_{WC}-conjugate

$\{A, T, G, C\}$, the Watson-Crick complementarity function θ_{WC}, and employ $u, v \in \{A, T, G, C\}^*$ for encoding a problem, then a possible intermolecular hybridization (see Fig. 3) may occur between v and u. To avoid such a structure at the time of encoding, if we use v, then we should refrain from utilizing any of its θ_{WC}-conjugates that are different from the word v. In addition to being relevant for DNA computing, the notion of θ-conjugates of a word is a generalization of conjugates of a word, a classical and well studied concept in combinatorics on words. These motivate us to explore various properties of θ-conjugates of a word and θ-conjugate-free languages when θ is an antimorphic involution.

Mahalingam et al. (2022) discussed several properties of the set of all θ-conjugates of a word when θ is an antimorphic involution. They mainly studied the cardinality of the set of all θ-conjugates of a word and occurrences of palindromes, θ-palindromes in the set of all θ-conjugates of a word. In this work, we further explore properties of the set of all θ-conjugates of a word when θ is an antimorphic involution. We also extend the idea of θ-conjugates of a word to θ-conjugates of a language L as $C_\theta(L) = \bigcup_{w \in L} C_\theta(w)$ and study various properties of $C_\theta(L)$ when θ is an antimorphic involution. The text is organized as follows. In Section 3, we study some properties of the set of all θ-conjugates of a word. We first investigate the structure of θ-conjugates of a word. Then, we discuss the solution of the equation $C_\theta(u) = C_\theta(v)$. Section 4 investigates the Chomsky hierarchy for $C_\theta(L)$ when L is regular, context-free, context-sensitive and recursively enumerable. In Section 5, we define and discuss the iterated θ-conjugate of words and languages. Section 6 investigates some decidability problems for θ-conjugate-free languages. We end our work with some concluding remarks.

2 Basic definitions and notations

An *alphabet* Σ is a finite non-empty set of letters. A *word* over Σ is a sequence of letters from Σ. The *length of a word* w, denoted by $|w|$, is the number of letters in w. A word with length zero is called the *empty word* and is denoted by λ. The set of all finite words over Σ is denoted by Σ^*, and $\Sigma^+ = \Sigma^* \setminus \{\lambda\}$. A *language* L over Σ is a subset of Σ^*. The set of all words over Σ of length n is denoted by Σ^n and the set of all words over Σ of length greater than or equal to n is denoted by $\Sigma^{\geq n}$. Let $Alph(w)$ denote the set of all letters that occur in a word w and $Alph(L) = \bigcup_{w \in L} Alph(w)$ where L is a language. A word u is a *factor* of a word w if $w = xuy$ for some $x, y \in \Sigma^*$. When $x = \lambda$ (respectively $y = \lambda$), u is called a *prefix* (respectively *suffix*) of w. If u is a prefix (respectively suffix) of w and $u \neq w$, then u is called a *proper prefix* (respectively *proper suffix*) of w. Set of all factors (respectively prefixes and suffixes) of a word w is denoted by $Fac(w)$ (respectively $Pref(w)$ and $Suf(w)$). For $x \in \Sigma^*$, $|w|_x$ denotes the number of occurrences of x in w. For a word $w = w_1 w_2 \cdots w_n$ where each $w_i \in \Sigma$, the *reversal* of w is $w^R = w_n \cdots w_2 w_1$. A word w is a *palindrome* if $w = w^R$. A word u is a *conjugate* of a word w if $u = yx$ and $w = xy$ for some $x, y \in \Sigma^*$ (see Figure 2). The set of all conjugates of w is denoted by $C(w)$. A word $w \in \Sigma^+$ is called *primitive* if $w = x^n$ implies $w = x$ and $n = 1$. For every $w \in \Sigma^+$, there exists a unique $\rho(w) \in \Sigma^+$, called the *root* of w, such that $\rho(w)$ is a primitive word and $w = \rho(w)^n$ for some integer $n \geq 1$.

A function $\theta : \Sigma^* \to \Sigma^*$ is said to be an *antimorphism* (respectively *morphism*) if $\theta(uv) = \theta(v)\theta(u)$ (respectively $\theta(uv) = \theta(u)\theta(v)$) for any $u, v \in \Sigma^*$. The function θ is called *involution* if θ^2 is the identity mapping on Σ^*. For a morphic/antimorphic involution θ, a word z is called a θ-*palindrome* if $z = \theta(z)$. For a morphic/antimorphic involution θ, a word u is a θ-*conjugate* of a word w if $uv = \theta(v)w$ for some $v \in \Sigma^*$. The set of all θ-conjugates of w is denoted by $C_\theta(w)$. Note that when θ is an antimorphic involution on $\{A, T, G, C\}^*$ such that $\theta(A) = T$, $\theta(T) = A$, $\theta(C) = G$, $\theta(G) = C$, then $\theta = \theta_{WC}$, i.e., θ is the *Watson-Crick complementarity function*. In this case $C_\theta(w)$ is called the set of all *Watson-Crick conjugates* of w.

In the rest of the text, we consider θ as an antimorphic involution over an alphabet Σ^*.

A *grammar* G is formally defined as a quadruple (V, T, P, S) where V is a finite set of non-terminal symbols, T is a finite set of terminal symbols, P is a finite set of production rules, and $S \in V$ is the start symbol. Using production rules of P, if a word $w \in T^*$ is derived from S, then we say w is generated by the grammar G, and it is denoted by $S \to^* w$. The set of all words generated by the grammar G is denoted by $L(G)$. A grammar G is said to be *context-free* if all the production rules of P are of the form $A \to u$, where $A \in V$ and $u \in (V \cup T)^*$. A grammar G is said to be *context-sensitive* if all the production rules of P are of the form $\alpha A \beta \to \alpha \gamma \beta$, where $A \in V$, $\alpha, \beta \in (V \cup T)^*$ and $\gamma \in (V \cup T)^+$. A grammar G is said to be *unrestricted* if all the production rules of P are of the form $u \to v$, where $u, v \in (V \cup T)^*$ with u containing at least one non-terminal symbol.

A *deterministic finite automaton* (DFA) M is defined by $M = (S', \Sigma, \delta, q_0, F)$, where S' is a finite set of states, Σ is an alphabet, $\delta : S' \times \Sigma \to S'$ is a transition function, $q_0 \in S'$ is the initial state and $F \subseteq S'$ is the set of all final states. A word is accepted by the DFA M if, starting from the initial state q_0, processing the word using the transition function δ leads to a state in F. The set of all words accepted by M is denoted by $L(M)$. A language L_1 is said to be *regular* if there exists a DFA M_1 such that $L_1 = L(M_1)$. A *linear bounded automaton* is a non-deterministic Turing machine $M' = (Q', \Sigma, \Gamma, \delta, q_0, <, >, B, F)$ where Q' is the set of all states, Γ is the tape alphabet, $\Sigma \subseteq \Gamma \setminus \{B\}$ is the input alphabet, $\delta : Q' \times \Gamma \to 2^{Q' \times \Gamma \times \{L, R, N\}}$ is the transition function, $B \in \Gamma$ is the blank symbol, $q_0 \in Q'$ is the initial state, $F \subseteq Q'$ is the set of

final states, $<\in \Sigma$ is the left end marker and $>\in \Sigma$ is the right end marker. Here, L, R and N denote the possible movements of the tape head: left, right, and no movement, respectively, as dictated by the transition function δ. A word w is accepted by the linear bounded automaton M' if the automaton, starting in its initial state q_0 with the input word $< w >$ on the tape, can transition through a sequence of states according to its transition function δ and eventually reach one of its final states in F. Here note that if w is a word accepted by a linear bounded automaton M', then w is in $(\Sigma \setminus \{<, >\})^*$. For a state q and two strings u, v over the tape alphabet Γ, we write uqv for the *configuration* of the Turing machine M' where the current state is q, the current tape content is uv, and the current head location is the first symbol of v. A move of the Turing machine M' from one configuration to another is denoted by $\vdash_{M'}$. The symbol $\vdash_{M'}^*$ represents an arbitrary number of moves of M'.

For all other concepts in formal language theory and combinatorics on words, the reader is referred to Linz (2012); Lothaire (1997); Rozenberg and Salomaa (Eds.); Shallit (2008).

We recall the following results from the literature, which will be used throughout our work.

Lemma 2.1. *Lyndon and Schützenberger (1962) Let $u, v, w \in \Sigma^+$.*

- *If $uv = vu$, then u and v are powers of a common primitive word.*

- *If $uv = vw$, then $u = xy$, $v = (xy)^k x$, $w = yx$ for some $k \geq 0$, $x \in \Sigma^+$ and $y \in \Sigma^*$.*

Proposition 2.2. *Kari and Mahalingam (2008b) For $u, v \in \Sigma^+$, if $uv = \theta(v)u$ and θ is an antimorphic involution, then $u = x(yx)^i$, $v = yx$ for some integer $i \geq 0$ and θ-palindromes $x \in \Sigma^+$, $y \in \Sigma^*$.*

3 Some Important Properties of set of all θ-conjugates of a Word

In this section, we investigate several combinatorial properties of the set of all θ-conjugates of a word. First, we examine the structure of elements of $C_\theta(w)$ for $w \in \Sigma^*$. This analysis will be useful later on. Next, we show that if $C_\theta(w)$ contains $|w| + 1$ elements, then $C_\theta(w)$ must contain a primitive word. Finally, we discuss the structure of words u and v such that $C_\theta(u) = C_\theta(v)$.

To discuss the structure of elements of $C_\theta(w)$ for some w, we first recall the following result from Kari and Mahalingam (2008b).

Proposition 3.1. *Kari and Mahalingam (2008b) For an antimorphic involution θ, consider u is a θ-conjugate of w, i.e., $uv = \theta(v)w$ for some $v \in \Sigma^*$. Then, either $u = xy$, $v = \theta(x)$ and $w = y\theta(x)$ for some $x, y \in \Sigma^*$, or $w = \theta(u)$ and $v = \beta w$ for some θ-palindrome β.*

Using Proposition 3.1, we obtain the following result, which will be used throughout this text.

Proposition 3.2. *For a word w, $C_\theta(w) = \{\theta(y)x : w = xy$ where $x, y \in \Sigma^*\}$.*

Proof: Consider $T = \{\theta(y)x : w = xy, x, y \in \Sigma^*\}$. If $u \in C_\theta(w)$, then $uv = \theta(v)w$ for some $v \in \Sigma^*$. Then by Proposition 3.1, $u = \theta(y)x$ and $w = xy$ for some $x, y \in \Sigma^*$ (see Figure 2). This implies $u \in T$, i.e., $C_\theta(w) \subseteq T$. Consider $z \in T$. Then, $z = \theta(y_1)x_1$ where $w = x_1 y_1$ for some $x_1, y_1 \in \Sigma^*$. If $x_1 = \lambda$, then for $v_1 = \beta y_1$ where β is a θ-palindrome, we have $zv_1 = \theta(y_1)\beta y_1 = \theta(v_1)w$, i.e., $z \in C_\theta(w)$. If $x_1 \in \Sigma^+$, then for $v_2 = y_1$, we have $zv_2 = \theta(y_1)x_1 y_1 = \theta(v_2)w$, i.e., $z \in C_\theta(w)$. Thus, $T \subseteq C_\theta(w)$. $\qquad\square$

In the following, we illustrate Proposition 3.2 with several examples. These examples also highlight an important property of the θ-conjugacy relation. Similar to the conjugacy relation, we define the θ-conjugacy relation R_θ on Σ^*, where $(u, v) \in R_\theta$ indicates that u is a θ-conjugate of v. It is known that the conjugacy relation is an equivalence relation Lothaire (1997). However, with the help of the following examples, we show that the θ-conjugacy relation is not an equivalence relation.

Example 3.3. *Consider $\Sigma = \{a, b, c, d\}$ and θ be such that $\theta(a) = b$, $\theta(c) = d$.*

1. *Consider $w = aac$, $u = daa$ and $v = dbb$. Then by Proposition 3.2, $C_\theta(w) = \{aac, daa, dba, dbb\}$, $C_\theta(u) = \{daa, bda, bbd, bbc\}$ and $C_\theta(v) = \{dbb, adb, aad, aac\}$. Now, $v \in C_\theta(w)$ and $w \in C_\theta(v)$. On the other hand, $u \in C_\theta(w)$ but $w \notin C_\theta(u)$. Thus, θ-conjugacy is not a symmetric relation.*

2. *Consider $w = abb$, $u = bbb$ and $v = bab$. Then by Proposition 3.2, $C_\theta(w) = \{abb, aab, aaa\}$, $C_\theta(u) = \{bbb, abb, aab, aaa\}$ and $C_\theta(v) = \{bab, aba, abb\}$. Now, $aab \in C_\theta(w)$, $w \in C_\theta(u)$ and $aab \in C_\theta(u)$. On the other hand, $aaa \in C_\theta(w)$, $w \in C_\theta(v)$ but $aaa \notin C_\theta(v)$. Thus, θ-conjugacy is not a transitive relation.*

By definition of conjugates, if $w \in \Sigma^+$ and $a \in Alph(w)$, then there exists an element in $C(w)$ that begins with a. However, this property does not hold for θ-conjugates, as illustrated in Example 3.3. In the following, we discuss the structure of elements of the set of all θ-conjugates of a word regarding this property.

Lemma 3.4. *For $u \in \Sigma^*$ and $a \in \Sigma$, $C_\theta(ua) = \{ua\} \cup \theta(a)C_\theta(u)$.*

Proof: We first show that $C_\theta(ua) \subseteq \{ua\} \cup \theta(a)C_\theta(u)$. Consider $x \in C_\theta(ua)$. Then for $ua = u'v'$, $x = \theta(v')u'$ where $u', v' \in \Sigma^*$. If $v' = \lambda$, then $x = ua$. Otherwise, if $v' \in \Sigma^+$, then $v' = za$ for some $z \in \Sigma^*$. This implies $x = \theta(a)\theta(z)u'$ and $u = u'z$. Then, $\theta(z)u' \in C_\theta(u)$. Therefore, $x \in \{ua\} \cup \theta(a)C_\theta(u)$ and $C_\theta(ua) \subseteq \{ua\} \cup \theta(a)C_\theta(u)$. Now, consider $y \in \{ua\} \cup \theta(a)C_\theta(u)$. If $y = ua$, then clearly $y \in C_\theta(ua)$. If $y \in \theta(a)C_\theta(u)$, then $y = \theta(a)\theta(q)p$ for $u = pq$, $p, q \in \Sigma^*$. This implies $y \in C_\theta(pqa)$, i.e., $y \in C_\theta(ua)$. Therefore, $\{ua\} \cup \theta(a)C_\theta(u) \subseteq C_\theta(ua)$. As a result, we have, $C_\theta(ua) = \{ua\} \cup \theta(a)C_\theta(u)$. □

Corollary 3.5. *For $v \in \Sigma^+$ and $u \in \Sigma^*$, $\theta(v)C_\theta(u) \subseteq C_\theta(uv)$.*

In addition, Proposition 2.2 and Lemma 3.4 lead us to the following.

Corollary 3.6. *For $u \in \Sigma^*$ and $a \in \Sigma$, $C_\theta(ua) = \theta(a)C_\theta(u)$ if and only if ua is a θ-palindrome.*

We provide some examples to illustrate Lemma 3.4 and Corollary 3.6.

Example 3.7. *Consider $\Sigma = \{a, b, c\}$ and θ be such that $\theta(a) = b$, $\theta(c) = c$.*

1. *If $w = bccb$, then $C_\theta(w) = \{bccb, abcc, acbc, accb, acca\}$. Note that $bccb$ is not a θ-palindrome. Now, $C_\theta(bcc) = \{bcc, cbc, ccb, cca\}$ and $C_\theta(bccb) = \{bccb\} \cup \theta(b)C_\theta(bcc)$.*

2. *If $w = abcab$, then $C_\theta(w) = \{abcab, aabca, ababc, abcaa\}$. Note that w is a θ-palindrome. Now, $C_\theta(abca) = \{abca, babc, bcab, bcaa\}$ and $C_\theta(abcab) = \theta(b)C_\theta(abca)$.*

For $w \in \Sigma^+$, $C(w)$ can contain at most $|w|$ elements and if $|C(w)| = |w|$, then each element of $C(w)$ is primitive. Mahalingam et al. (2022) proved that the set of all θ-conjugates of a word w can have at most $|w| + 1$ elements. We now prove that for $w \in \Sigma^*$, if $|C_\theta(w)| = |w| + 1$, then $C_\theta(w)$ must contain at least one primitive word. To prove this we need the following result.

Lemma 3.8. *For $a \in \Sigma$ and $w \in \Sigma^n$ with $n \geq 2$, if each element of $C_\theta(w)$ is non-primitive and $a^n \in C_\theta(w)$, then $w = a^n$ with $\theta(a) = a$.*

Proof: Consider each element of $C_\theta(w)$ is non-primitive and $a^n \in C_\theta(w)$. Then for some $x, y \in \Sigma^*$, $\theta(y)x = a^n$ and $w = xy$. This implies $\theta(y) = a^j$ and $x = a^i$ for some $i, j \geq 0$ with $i + j = n$. Then, $w = xy = a^i\theta(a)^j$. Let us assume that $a \neq \theta(a)$. If $j = 0$, then $w = a^n$ and $\theta(a)a^{n-1} \in C_\theta(w)$ is a primitive word, which is a contradiction. Similarly, if $i = 0$, then $w = \theta(a)^n$ and $a\theta(a)^{n-1} \in C_\theta(w)$ is a primitive word, which is a contradiction. For $i, j \geq 1$, w is itself a primitive word, which also leads to a contradiction. Therefore, $\theta(a) = a$ and $w = a^n$. $\qquad\square$

Using Lemma 3.8, we now prove the following.

Proposition 3.9. *If $|C_\theta(w)| = n + 1$ for a word w of length $n \geq 1$, then there exists a primitive word in $C_\theta(w)$.*

Proof: For a word w of length $n \geq 1$, consider $|C_\theta(w)| = n + 1$.

- For $n = 1$, the word w is a letter, i.e., a primitive word. Since $w \in C_\theta(w)$, $C_\theta(w)$ contains a primitive word.

- For $n \geq 2$, let us assume that each element of $C_\theta(w)$ is non-primitive. Now, the length of the root of each element in $C_\theta(w)$ is at most $\lfloor \frac{|w|}{2} \rfloor$. Consider $l = \lfloor \frac{|w|}{2} \rfloor$ and $w = w_1 w_2 \cdots w_l w_{l+1} \cdots w_n$ where each $w_j \in \Sigma$ for $1 \leq j \leq n$. For each i ($0 \leq i \leq l - 1$), let $\alpha_i = \beta_i \gamma_i$ where $\beta_i = \theta(w_{l-i} \cdots w_n)$ and $\gamma_i = w_1 w_2 \cdots w_{l-i-1}$. Then for each i ($0 \leq i \leq l - 1$), $|\beta_i| > l$ and $\alpha_i \in C_\theta(w)$. Consider $\rho(\alpha_i) = z_i$ for each i ($0 \leq i \leq l - 1$). Then, $|z_i| \leq l$ for each i ($0 \leq i \leq l - 1$). Now, each $\alpha_0, \alpha_1, \cdots, \alpha_{l-1}$ has a common prefix β_0 where $\beta_0 = \theta(w_n)\theta(w_{n-1}) \cdots \theta(w_l)$ and $|\beta_0| > l$. This implies each z_i is a proper prefix of β_0 as $|z_i| \leq l$ and $|\beta_0| > l$ where $0 \leq i \leq l - 1$. Since each element of $C_\theta(w)$ is non-primitive and $|C_\theta(w)| = |w| + 1$, each z_i ($0 \leq i \leq l - 1$) is a distinct prefix of β_0. Then, the possible lengths of z_i's are $1, 2, \cdots, l$.

  For $i_1 \neq i_2$ ($0 \leq i_1, i_2 \leq l - 1$), if the lengths of z_{i_1} and z_{i_2} are equal, then as z_{i_1} and z_{i_2} are prefixes of β_0, we have $z_{i_1} = z_{i_2}$, i.e., $\alpha_{i_1} = \alpha_{i_2}$, which is a contradiction to our assumption that $|C_\theta(w)| = |w| + 1$.

  Otherwise, consider the length of each z_i is distinct. Since the length of the root of each element in $C_\theta(w)$ is at most l and there are l distinct elements $\alpha_0, \alpha_1, \ldots, \alpha_{l-1}$ in $C_\theta(w)$ with a commom prefix β_0, we have $|\rho(\alpha_{i_3})| = 1$ for some $0 \leq i_3 \leq l - 1$. Then by Lemma 3.8, $w = a^n$ with $\theta(a) = a$, which is a contradiction to our assumption that $|C_\theta(w)| = |w| + 1$.

Thus, if $|C_\theta(w)| = |w| + 1$, then there exists a primitive word in $C_\theta(w)$. $\qquad\square$

For $u, v \in \Sigma^*$, the equality of $C_\theta(u)$ and $C_\theta(v)$ was studied in Mahalingam et al. (2022), and the following result was proved.

Theorem 3.10. *Mahalingam et al. (2022) Let $u, v \in \Sigma^*$ such that $C_\theta(u) = C_\theta(v)$.*

1. *If $|C_\theta(u)| = 1$, then $v = u$.*

2. *If $|C_\theta(u)| = 2$, then $v = u$ or $v = \theta(u)$.*

3. *If $|C_\theta(u)| \geq 3$, then one of the following holds:*

   (a) *$u = v$*

   (b) *$u = a^{n_1} u' a^{n_2}$ and $v = u^R$ where $a \in \Sigma$, $u' \in \Sigma^+$, $a \notin Pref(u')$, $a \notin Suf(u')$, $u' = u'^R$, $n_1, n_2 \geq 1$, $n_1 \neq n_2$, and $\theta(c) = c \, \forall c \in Alph(u)$.*

Remark 3.11. *Mahalingam et al. (2022) proved that if $C_\theta(u) = C_\theta(v)$ and $|C_\theta(u)| \geq 3$ for $u, v \in \Sigma^*$, then either $u = v$ or $u = v^R$ (Proposition 3.15 in Mahalingam et al. (2022)). However, the detailed structures of u and v were provided within the proof of Proposition 3.15 in Mahalingam et al. (2022). Since we will later use these structural details, we have explicitly included them in Theorem 3.10. Additionally, to ensure clarity and verifiability, we have added the proof of these structural details in the appendix.*

With the help of a few examples, we now show that the converse of Theorem 3.10 is not true.

Example 3.12. *Let $\Sigma = \{a, b, c, d\}$ and θ be such that $\theta(a) = b$, $\theta(c) = c$ and $\theta(d) = d$.*

1. *Consider $u = ca$ and $v = \theta(u) = bc$. Then, $|C_\theta(u)| = 2$ and $cb \in C_\theta(v) \setminus C_\theta(u)$.*

2. *Consider $u = cdccdc^2$ and $v = u^R = c^2 dccdc$. Then, $|C_\theta(u)| \geq 3$ and $cdcccdc \in C_\theta(v) \setminus C_\theta(u)$.*

Hence, in both cases, $C_\theta(u) \neq C_\theta(v)$.

We now discuss necessary and sufficient conditions for the equality of $C_\theta(u)$ and $C_\theta(v)$ for words u and v. Since $u = v$ always implies $C_\theta(u) = C_\theta(v)$, we focus on characterizing words u and v such that $u \neq v$ and $C_\theta(u) = C_\theta(v)$. Furthermore, as $C_\theta(u) = C_\theta(v)$ and $|C_\theta(u)| = 1$ implies $u = v$ (by Theorem 3.10, Assertion(1)), we characterize pairs $u, v \in \Sigma^*$ satisfying $u \neq v$, $C_\theta(u) = C_\theta(v)$, and $|C_\theta(u)| = |C_\theta(v)| > 1$.

Proposition 3.13. *Consider u and v are two distinct words of same length k such that $|C_\theta(u)| = |C_\theta(v)| = 2$. Then, $C_\theta(u) = C_\theta(v)$ if and only if u and v satisfy one of the following:*

1. *$u = c$ and $v = \theta(c)$ for some $c \in \Sigma$ with $\theta(c) \neq c$.*

2. *$u = ab$ and $v = ba$ for some $a, b \in \Sigma$ with $a \neq b$, $\theta(a) = a$, and $\theta(b) = b$.*

Proof: For $u, v \in \Sigma^k$, $u \neq v$ and $|C_\theta(u)| = |C_\theta(v)| = 2$, consider $C_\theta(u) = C_\theta(v)$. Then by Theorem 3.10, $v = \theta(u)$ where $\theta(u) \neq u$.

- Consider $|u| = k = 1$. Then as $|C_\theta(u)| = 2$, $u = c$ and $v = \theta(c)$ for some $c \in \Sigma$ with $\theta(c) \neq c$.

- Consider $|u| = k \geq 2$. Since $|C_\theta(u)| = 2$ and $u \neq v$, we have the following: for each $\alpha, \beta \in \Sigma^+$ for which $u = \alpha\beta$, $\theta(\beta)\alpha \in C_\theta(u)$ must be equal to either $u = \alpha\beta$ or $v = \theta(u) = \theta(\beta)\theta(\alpha)$. If $\theta(\beta)\alpha = u$, then by Proposition 2.2, u is a θ-palindrome, which is a contradiction to the fact that $\theta(u) \neq u$. Thus, $\theta(\beta)\alpha = \theta(u)$, i.e., $\theta(\beta)\alpha = \theta(\beta)\theta(\alpha)$, i.e., $\theta(\alpha) = \alpha$.

  Now, for each $\alpha, \beta \in \Sigma^+$ for which $u = \alpha\beta$, we have $\alpha\theta(\beta) \in C_\theta(v)$ (since $v = \theta(u) = \theta(\beta)\theta(\alpha)$). Since $C_\theta(u) = C_\theta(v)$ and $|C_\theta(u)| = 2$, $\alpha\theta(\beta)$ is equal to either u or v. If $\alpha\theta(\beta) = v$,

i.e., $\alpha\theta(\beta) = \theta(\beta)\theta(\alpha)$, then by Proposition 2.2, v is a θ-palindrome, which is a contradiction to the fact that $\theta(u) \neq u$. Thus, $\alpha\theta(\beta) = u$, i.e., $\alpha\theta(\beta) = \alpha\beta$, i.e., $\theta(\beta) = \beta$.

Therefore, for each $\alpha, \beta \in \Sigma^+$ satisfying $u = \alpha\beta$, we have $\theta(\alpha) = \alpha$, $\theta(\beta) = \beta$.

- If $|u| = k = 2$, then $u = ab$ and $v = ba$ for some $a, b \in \Sigma$ with $\theta(a) = a$, $\theta(b) = b$. Here, $a \neq b$, otherwise $u = v = a^2$, which contradicts the fact that $u \neq v$.

- If $|u| = k \geq 3$, then as each non-empty proper prefix and suffix of u are θ-palindromes, $u = v = d^k$ for some $d \in \Sigma$ with $\theta(d) = d$, which is a contradiction to the fact that $u \neq v$.

The converse is straightforward. $\qquad\square$

Proposition 3.14. *Consider u and v are two distinct words of same length such that $|C_\theta(u)| = |C_\theta(v)| \geq 3$. Then, $C_\theta(u) = C_\theta(v)$ if and only if $u = a^{m+1}ba^m$ and $v = a^m ba^{m+1}$ where $m \geq 1$, $a, b \in \Sigma$, $a \neq b$, $\theta(a) = a$ and $\theta(b) = b$.*

Proof: For $u, v \in \Sigma^*$, $u \neq v$ and $|C_\theta(u)| = |C_\theta(v)| \geq 3$, consider $C_\theta(u) = C_\theta(v)$. Then, by Theorem 3.10,
$$u = a^{n_1}u'a^{n_2} \text{ and } v = a^{n_2}u'a^{n_1}$$
where $a \in \Sigma$, $n_1, n_2 \geq 1$, $n_1 \neq n_2$, $u' \in \Sigma^+$, $a \notin Pref(u')$, $a \notin Suf(u')$, $u' = u'^R$ and $\theta(c) = c \ \forall c \in Alph(u)$.

Without loss of generality, let us take $n_1 > n_2$. If $|u'| \geq 2$, then according to the above conditions, we have $u' = d_1\alpha d_1$ for some $d_1 \in \Sigma$ and $\alpha \in \Sigma^*$ with $d_1 \neq a$ and $\theta(d_1) = d_1$. This implies $u = a^{n_1}d_1\alpha d_1 a^{n_2}$ and $v = a^{n_2}d_1\alpha d_1 a^{n_1}$ where $n_1 > n_2$. Then, $a^{n_1}d_1\theta(\alpha)a^{n_2}d_1 \in C_\theta(v) \setminus C_\theta(u)$, which is a contradiction to our assumption that $C_\theta(u) = C_\theta(v)$. Thus as $u' \in \Sigma^+$, $|u'| = 1$. Then, $u = a^{n_1}ba^{n_2}$ and $v = a^{n_2}ba^{n_1}$ where $u' = b \in \Sigma$, $a \neq b$ and $\theta(b) = b$. If $n_1 - n_2 \geq 2$, i.e., $n_1 - n_2 = 2 + k$ for $k \geq 0$, then $u = a^{n_2+k+2}ba^{n_2}$ and $v = a^{n_2}ba^{n_2+k+2}$. Now, $a^{k+n_2+1}ba^{n_2+1} \in C_\theta(v) \setminus C_\theta(u)$, which is a contradiction to our assumption that $C_\theta(u) = C_\theta(v)$. Thus, as $n_1 > n_2$, $n_1 - n_2 = 1$. Then, $u = a^{n_2+1}ba^{n_2}$ and $v = a^{n_2}ba^{n_2+1}$.

Conversely, let $u = a^{m+1}ba^m$ and $v = a^m ba^{m+1}$ where $m \geq 1$, $a, b \in \Sigma$, $\theta(a) = a$, $\theta(b) = b$ and $a \neq b$. Then $C_\theta(u) = \{a^{m+1+j}ba^{m-j} : 0 \leq j \leq m\} \cup \{v\}$, and $C_\theta(v) = \{v\} \cup \{a^{m+i}ba^{m+1-i} : 1 \leq i \leq m+1\} \cup \{a^{m+1}ba^m\}$. Since $a^{m+1}ba^m \in \{a^{m+i}ba^{m+1-i} : 1 \leq i \leq m+1\}$, we have $C_\theta(v) = \{v\} \cup \{a^{m+i}ba^{m+1-i} : 1 \leq i \leq m+1\}$. Consider $i = i'+1$. Then, $1 \leq i \leq m+1$ implies $0 \leq i' \leq m$. Thus, $C_\theta(v) = \{v\} \cup \{a^{m+i'+1}ba^{m-i'} : 0 \leq i' \leq m\}$. Therefore, $C_\theta(u) = C_\theta(v)$. $\qquad\square$

It is well known that for given $u, v \in \Sigma^*$, $C(u) = C(v)$ if and only if $u \in C(v)$. But this statement is not true in general for θ-conjugates of a word. Propositions 3.13 and 3.14 describe necessary and sufficient conditions for the equality of $C_\theta(u)$ and $C_\theta(v)$ for distinct words u and v with $|C_\theta(u)| = |C_\theta(v)| \geq 2$. Also, it is clear from Propositions 3.13 and 3.14 that if sets of all θ-conjugates of two distinct words $u \in \Sigma^{\geq 2}$ and $v \in \Sigma^{\geq 2}$ are equal, then u, v must be binary words and $\theta(c) = c$ for all $c \in Alph(u) \cup Alph(v)$. Now, combining Propositions 3.13 and 3.14, we have the following characterizations of u and v such that sets of all θ-conjugates of u and v are equal.

Theorem 3.15. *For $u, v \in \Sigma^*$, $C_\theta(u) = C_\theta(v)$ if and only if one of the following holds:*

1. $u = v$.

2. *$u = ab$ and $v = ba$ for $a, b \in \Sigma$ with $\theta(a) = a$, $\theta(b) = b$ and $a \neq b$.*

3. *$u = c$ and $v = \theta(c)$ for $c \in \Sigma$ with $\theta(c) \neq c$.*

4. *$u = a^{m+1}ba^m$ and $v = a^m b a^{m+1}$ for $a, b \in \Sigma$ with $\theta(a) = a$, $\theta(b) = b$, $a \neq b$ and $m \geq 1$.*

4 θ-conjugates of a Language And Closure Properties of θ-conjugate Operation

Conjugate operation or cyclic shift operation (Jirásková and Okhotin (2008)) is a unary operation on formal languages defined as $C(L) = \bigcup_{w \in L} C(w)$ for some $L \subseteq \Sigma^*$. Here, we call $C(L)$ as the set of all conjugates of the language L. Jirásková and Okhotin (2008) showed that the family of regular languages is closed under the conjugate operation. In addition, Oshiba (1972) demonstrated that if L is a context-free language, then $C(L)$ is also context-free. Families of context-sensitive and recursively enumerable languages have also been shown to be closed under the conjugate operation by Brandstädt (1981). Now, the definition of θ-conjugate of a word can be naturally extended to a language. In this section, we first define the θ-conjugate operation on languages and investigate the closure property of certain families of languages under the θ-conjugate operation.

Definition 4.1. *θ-conjugate operation is a unary operation on formal languages defined as*

$$C_\theta(L) = \bigcup_{w \in L} C_\theta(w)$$

for some $L \subseteq \Sigma^*$. Here, we call $C_\theta(L)$ as the set of all θ-conjugates of the language L.

We now investigate closure properties of certain well-known families of languages with respect to θ-conjugate operation. For this, we need the following result whose proof is straightforward.

Lemma 4.2. *For a morphic involution ψ and an antimorphic involution θ over Σ^*, if $\psi(a) = \theta(a) \ \forall a \in \Sigma$, then $\psi(u) = \theta(u^R)$ for $u \in \Sigma^*$.*

First, we show that the family of regular languages is closed under θ-conjugate operation. The proof is very similar to the argument presented by Jirásková and Okhotin (2008) on the closure of regular languages under the conjugate operation.

Theorem 4.3. *If $L \subseteq \Sigma^*$ is regular, then $C_\theta(L)$ is also regular.*

Proof: Consider a deterministic finite automaton (DFA) A which accepts L. Let Q', q_0, and F be the set of all states, the initial state, and the set of all final states of A, respectively. Now, for each state $q \in Q'$, we first construct two DFAs C_q and B_q with same states and same transition functions as A where q_0 is the initial state of C_q, $\{q\}$ is the set of all final states of C_q, q is the initial state of B_q and F is the set of all final states of B_q. Consider a morphic involution $\psi : \Sigma^* \to \Sigma^*$ such that $\psi(a) = \theta(a)$ for all $a \in \Sigma$. Then, as the family of regular languages is closed under reversal and morphism Lothaire (1997), there exists a DFA B_q' which accepts $\psi((L(B_q))^R)$, i.e., $\theta(L(B_q))$ (by Lemma 4.2). Now, by definition, a string w is in $C_\theta(L)$ if and only if $w = \theta(v)u$ for $uv \in L$. Then, in the accepting computation of A on uv, there exists an intermediate state q of A such that the computation of A on u ends in the state q, while v is accepted by A starting from q. Then, $\theta(v)$ is accepted by B_q' and u is accepted by C_q and

$C_\theta(L) = \bigcup_{q \in Q'} L(B'_q)L(C_q)$. Since the family of regular languages is closed under concatenation and union Lothaire (1997), $C_\theta(L)$ is regular. $\square$

We now consider the closure property of family of context-free languages under θ-conjugate operation. To proceed, we require the following result, commonly known as the *pumping lemma for Context-Free Languages*:

Theorem 4.4. *Shallit (2008) If L is context-free, then there exists a constant n such that for all $z \in L$ with $|z| \geq n$, there exists a decomposition $z = uvwxy$ with $|vwx| \leq n$ and $|vx| \geq 1$ such that for all $i \geq 0$, we have $uv^iwx^iy \in L$.*

Theorem 4.5. *The family of context-free languages is not closed under θ-conjugate operation.*

Proof: Consider the following context-free language $L = \{a^nb^kc^kd^n \mid n,k \geq 1\}$ over $\Sigma = \{a,b,c,d\}$. Suppose $\theta(e) = e$ for all $e \in \Sigma$. Then, $C_\theta(L) = \{d^ia^nb^kc^kd^{n-i} \mid n,k \geq 1, 0 \leq i \leq n\} \cup \{d^nc^ja^nb^kc^{k-j} \mid n,k \geq 1, 0 < j \leq k\} \cup \{d^nc^kb^la^nb^{k-l} \mid n,k \geq 1, 0 < l \leq k\}$. Consider the regular language $L' = \{d^{n_1}c^{m_1}a^{m_2}b^{n_2} \mid n_1, m_1,$
$m_2, n_2 \geq 1\}$. Now, if $C_\theta(L)$ is context-free, then as the intersection of a context-free language and a regular language is context-free Shallit (2008), $C_\theta(L) \cap L' = \{d^nc^ka^nb^k \mid n,k \geq 1\}$ is context-free. But by pumping lemma for context-free languages (Theorem 4.4), $\{d^nc^ka^nb^k \mid n,k \geq 1\}$ is not context-free (proof is in Appendix), which is a contradiction. Thus, $C_\theta(L)$ is not context-free. $\square$

We, furthermore, consider the closure property of the family of context-sensitive languages and show that for a given context sensitive language L_1, $C_\theta(L_1)$ is also context sensitive by constructing a linear bounded automaton which accepts $C_\theta(L_1)$. In the subsequent discussion, we will establish this result for the alphabet $\Sigma = \{a,b\}$.

Theorem 4.6. *If $L_1 \subseteq \{a,b\}^+$ is context-sensitive, then $C_\theta(L_1)$ is also context-sensitive.*

Proof: Let $L_1 \subseteq \{a,b\}^+$ be a context-sensitive language. Since the family of context-sensitive languages is closed under reversal Rozenberg and Salomaa (Eds.), L_1^R is also context-sensitive. For $\Sigma = \{a,b\}$, let $\psi : \Sigma^* \to \Sigma^*$ be a morphic involution such that $\psi(c) = \theta(c)$ for all $c \in \Sigma$. Consider

$$L' = \bigcup_{uv \in L_1^R} \{\psi(u)v^R\}.$$

Then, by Lemma 4.2,

$$L' = \bigcup_{uv \in L_1^R} \{\psi(u)v^R\} = \bigcup_{v^Ru^R \in L_1} \{\theta(u^R)v^R\} = C_\theta(L_1).$$

Now, as L_1^R is a context-sensitive language, there exists a linear bounded automaton $M = (Q', \Sigma \cup \{<,>\}, \Gamma, \delta, q'_0, <, >, B, F)$ that accepts L_1^R (Linz (2012)) and uses $(k+2) \cdot |z| + 2$ tape cells for any $z \in \Sigma^+$ with the initial configuration $q'_0 < \underbrace{BB \cdots B}_{((|z|-1) \text{ no. of blank cells})} z \underbrace{BB \cdots B}_{((k \cdot |z|+1) \text{ no. of blank cells})} >$ where k is a constant.

Using M, we now construct a linear bounded automaton M' which accepts L'. Let $M' = (Q'', \Sigma \cup \{<,>\}, \Gamma', \delta', q_0, <, >, B, F)$ where $Q'' = Q' \cup \{q_i \mid 0 \leq i \leq 15\}$, $\Gamma' = \Gamma \cup \{\#, X\}$, $\delta'(q,c) = \delta(q,c)$ for all $q \in Q'$, $c \in \Gamma$. For M', we will use $(k+2) \cdot |w| + 2$ tape cells for each $w \in \Sigma^*$. Also, for any
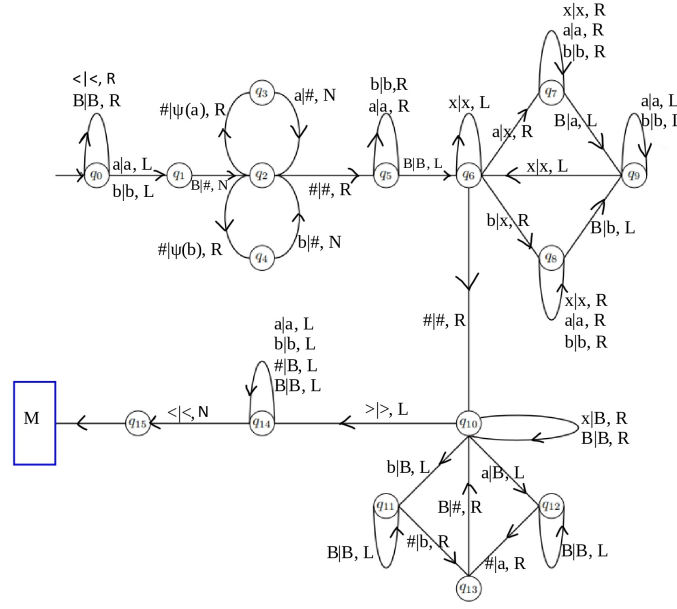
Fig. 4: Transition diagram for M'

$w \in \Sigma^*$, we consider the initial configuration of M' as $q_0 < \underset{(|w| \text{ no. of blank cells})}{BB \cdots B} w \underset{(k \cdot |w| \text{ no. of blank cells})}{BB \cdots B} >$. In
the transition diagram (Figure 4), we have discussed the transition rules of M' in details. Here we have
assumed that $q_0' = q_{15}$. Also, in Figure 4, the label $a_1|b_1, R$ (resp. L or N) on the arc from state q_i to
state q_j signifies that when the machine is in state q_i with the head reading a_1, the machine goes to state
q_j, writes b_1, and moves the head to the right (resp. left or stay in its current position).

Suppose $\alpha \in \Sigma^+$ is an input to M'. Then, for every factorization xy of α where $x, y \in \Sigma^*$, the
linear bounded automaton M' computes the following: $q_0 < \underset{(|\alpha| \text{ no. of blank cells})}{BB \cdots B} xy \underset{(k \cdot |\alpha| \text{ no. of blank cells})}{BB \cdots B} >$
$\vdash_{M'}^* q_{15} < \underset{(|\alpha|-1 \text{ no. of blank cells})}{BB \cdots B} \psi(x)y^R \underset{(k \cdot |\alpha|+1 \text{ no. of blank cells})}{BB \cdots B} >$, which is the initial configuration of M
for $\psi(x)y^R$. If, for at least one such factorization x_1y_1 of α, $\psi(x_1)y_1^R$ is accepted by M, then M' accepts
α. Conversely, if none of these factorizations lead to acceptance by M, then M' rejects α.

Now, we show that our machine M' accepts only elements of L'. Consider $\beta \in L'$. Then, for some
$u_1v_1 \in L_1^R$, $\beta = \psi(u_1)v_1^R$. If β is an input to M', then as ψ is an involution, M' computes the following:
$q_0 < BB \cdots B\psi(u_1)v_1^R BB \cdots B > \vdash_{M'}^* q_{15} < BB \cdots Bu_1v_1BB \cdots B >$. Since, $u_1v_1 \in L_1^R$, M
accepts u_1v_1. Thus, M' accepts β.

Now, consider $\gamma \notin L'$ be an input in M'. Then, for every factorization $\gamma_1\gamma_2$ of γ where $\gamma_1, \gamma_2 \in \Sigma^*$,
the linear bounded automaton M' computes the following:
$q_0 < BB \cdots B\gamma_1\gamma_2BB \cdots B > \vdash_{M'}^* q_{15} < BB \cdots B\psi(\gamma_1)\gamma_2^RBB \cdots B >$. If, for at least one such

factorization $\gamma_1' \gamma_2'$ of γ, the input $\psi(\gamma_1') \gamma_2'^R$ is accepted by M, then $\psi(\gamma_1') \gamma_2'^R \in L_1^R$. This implies $\gamma_1' \gamma_2' \in L'$, which is a contradiction as $\gamma = \gamma_1' \gamma_2' \notin L'$. Thus, for every factorization $\gamma_1 \gamma_2$ of γ where $\gamma_1, \gamma_2 \in \Sigma^*$, the linear bounded automaton M rejects $\psi(\gamma_1) \gamma_2^R$. Therefore, M' rejects γ. $\square$

Similar to Theorem 4.6, we can prove the following result for a context-sensitive language L over an arbitrary alphabet Σ.

Theorem 4.7. *If $L \subseteq \Sigma^+$ is context-sensitive language, then $C_\theta(L)$ is also context-sensitive.*

We now show that if L is a recursively enumerable language, then $C_\theta(L)$ is also recursively enumerable by constructing an unrestricted grammar that generates $C_\theta(L)$. This construction is simple but lengthy. So, we include this construction in the Appendix.

Theorem 4.8. *If L is a recursively enumerable language, then $C_\theta(L)$ is also recursively enumerable.*

5 Iterated θ-conjugate of Words And Languages

In this section, we first define the iterated θ-conjugate of a word and language. Then we characterize the elements of iterated θ-conjugate of a word. We also count the number of elements in the iterated θ-conjugate of a word. Then we study the closure property of iterated θ-conjugate operation. Throughout this section, we use $|u|_{a,\theta(a)}$ to represent the combined count of letters a and $\theta(a)$ in the string $u \in \Sigma^*$, i.e.,

$$|u|_{a,\theta(a)} = \begin{cases} |u|_a + |u|_{\theta(a)} & \text{if } \theta(a) \neq a \\ |u|_a & \text{if } \theta(a) = a. \end{cases}$$

Definition 5.1. *For a language L, we define $C_\theta{}^n(L)$ recursively as $C_\theta{}^0(L) = L$ and $C_\theta{}^n(L) = C_\theta{}^{n-1}(C_\theta(L))$. For a word w, we define $C_\theta{}^n(w) = C_\theta{}^n(\{w\})$, $n \geq 0$.*

Finally, we define the iterated θ-conjugate of a word w and a language L as $C_\theta{}^*(w) = \bigcup_{n \geq 0} C_\theta{}^n(w)$ and $C_\theta{}^*(L) = \bigcup_{n \geq 0} C_\theta{}^n(L)$, respectively.

As $w \in C_\theta(w)$ for any $w \in \Sigma^*$, we have the following.

Lemma 5.2. *For $w \in \Sigma^*$ and $m, k \geq 0$, $C_\theta{}^m(w) \subseteq C_\theta{}^{m+k}(w)$.*

Since the conjugacy relation is an equivalence relation Lothaire (1997), $u \in C(w)$ implies $C(u) = C(w)$. Then, $C(C(w)) = C^2(w) = C(w), C(C^2(w)) = C^3(w) = C(w), \ldots, C(C^{i-1}(w)) = C^i(w) = C(w)$ for all $i \geq 1$. But this is not true for θ-conjugates of a word. For an example, $ba \in C_\theta{}^2(ab)$ but $ba \notin C_\theta(ab)$ where $\theta(a) = b$. We now show that for a given $w \in \Sigma^n$, there exists an integer m depending on n such that $C_\theta{}^m(w)$ contains all words u such that $|u|_{a,\theta(a)}$ is equal to $|w|_{a,\theta(a)}$ for all $a \in \Sigma$. To prove this, we need the following results.

From the definition of θ-conjugates, it is clear that if u is an element of $C_\theta{}^m(w)$ for some $m \geq 1$, then for all $a \in \Sigma$, the combined count of letters a and $\theta(a)$ in both u and w must be equal.

Lemma 5.3. *If $u \in C_\theta{}^m(w)$ for any $m \geq 1$, then $|u|_{a,\theta(a)} = |w|_{a,\theta(a)}$ for all $a \in \Sigma$.*

Now, we prove the following.

Lemma 5.4. *Consider $u, w \in \Sigma^+$, $z \in \Sigma^*$, $|u| \geq 3$, and $a \in \Sigma$. If $uz \in C_\theta{}^m(w)$ where $m \geq 1$ and $|u|_{a,\theta(a)} \geq 1$, then $u'za \in C_\theta{}^{m+4}(w)$ for some $u' \in \Sigma^+$.*

Proof: Let $u \in \Sigma^+$ and $uz \in C_\theta{}^m(w)$ where $|u|_{a,\theta(a)} \geq 1$. We now have the following cases.

- If $|u|_a \geq 1$, then for some $\alpha, \beta \in \Sigma^*$, $u = \alpha a \beta$ and

$$
\begin{aligned}
\alpha a \beta z \in C_\theta{}^m(w) &\implies \theta(z)\theta(\beta)\alpha a \in C_\theta{}^{m+1}(w) \\
&\implies \theta(a)\theta(z)\theta(\beta)\alpha \in C_\theta{}^{m+2}(w) \\
&\implies \theta(\alpha)\beta z a \in C_\theta{}^{m+3}(w) \\
&\implies \theta(\alpha)\beta z a \in C_\theta{}^{m+4}(w) \qquad \text{(by Lemma 5.2)}
\end{aligned}
$$

- If $|u|_{\theta(a)} \geq 1$, then for some $\alpha', \beta' \in \Sigma^*$, $u = \alpha'\theta(a)\beta'$ and

$$
\begin{aligned}
\alpha'\theta(a)\beta' z \in C_\theta{}^m(w) &\implies \theta(z)\theta(\beta')a\alpha' \in C_\theta{}^{m+1}(w) \\
&\implies \theta(\alpha')\theta(a)\theta(z)\theta(\beta') \in C_\theta{}^{m+2}(w) \\
&\implies \beta' z a\alpha' \in C_\theta{}^{m+3}(w) \\
&\implies \theta(\alpha')\beta' z a \in C_\theta{}^{m+4}(w).
\end{aligned}
$$

Hence, we have the desired result. $\qquad\qquad\square$

For any $v \in \Sigma^2$ and $a, b \in \Sigma$, if $|v|_{a,\theta(a)} \geq 1$ and $|v|_{b,\theta(b)} \geq 1$, then the following can be shown using a similar proof technique of Lemma 5.4.

Lemma 5.5. *Consider* $v, z, w \in \Sigma^+$, $|v| = 2$ *and* $a, b \in \Sigma$. *If* $vz \in C_\theta{}^m(w)$ *where* $m \geq 1$, $|v|_{a,\theta(a)} \geq 1$ *and* $|v|_{b,\theta(b)} \geq 1$, *then* $zab \in C_\theta{}^{m+4}(w)$.

With the help of Lemmas 5.3, 5.4, and 5.5, we now prove that if w is a word of length at least three, then for all integer $m \geq 4|w| - 6$, $C_\theta{}^m(w)$ is equal to the collection of all strings u such that $|u|_{a,\theta(a)}$ is equal to $|w|_{a,\theta(a)}$ for all $a \in \Sigma$.

Proposition 5.6. *If* $w \in \Sigma^n$ *with* $n \geq 3$, *then for all* $m \geq 4n - 6$, $C_\theta{}^m(w) = \{u \mid |u|_{a,\theta(a)} = |w|_{a,\theta(a)} \ \forall a \in \Sigma\}$.

Proof: Let w be a word of length $n \geq 3$ and $T = \{u \mid |u|_{a,\theta(a)} = |w|_{a,\theta(a)} \ \forall a \in \Sigma\}$. By Lemma 5.3, we know that $C_\theta{}^{m_1}(w) \subseteq T$ for any $m_1 \geq 1$. Then, $C_\theta{}^m(w) \subseteq T$ for all $m \geq 4n - 6$. Now, we prove that $T \subseteq C_\theta{}^m(w)$ for all $m \geq 4n - 6$. Consider $x \in T$. Then, as $|x|_{a,\theta(a)} = |w|_{a,\theta(a)}$ for all $a \in \Sigma$, we have, $|x| = |w| = n \geq 3$. Let $x = x_1 x_2 \cdots x_n$ where each $x_j \in \Sigma$ for $1 \leq j \leq n$. Since $|x|_{x_i,\theta(x_i)} = |w|_{x_i,\theta(x_i)}$ for each $i (1 \leq i \leq n)$, it follows that w must contain at least one occurrence of x_i or $\theta(x_i)$ for each $i (1 \leq i \leq n)$. Then, for each i $(1 \leq i \leq n)$, w can be written as $w = \alpha_i x_i \beta_i$ or $w = \alpha_i \theta(x_i)\beta_i$ for some $\alpha_i, \beta_i \in \Sigma^*$.

If $w = \alpha_1 x_1 \beta_1$, then $\theta(\beta_1)\alpha_1 x_1 \in C_\theta(w)$. Since for each i $(2 \leq i \leq n - 2)$, either $|\theta(\beta_1)\alpha_1|_{x_i} \geq 1$ or $|\theta(\beta_1)\alpha_1|_{\theta(x_i)} \geq 1$, by Lemma 5.4, we have $vx_1 x_2 \cdots x_{n-2} \in C_\theta{}^{1+4(n-3)}(w)$, i.e., $vx_1 x_2 \cdots x_{n-2} \in C_\theta{}^{4n-11}(w)$ for some $v \in \Sigma^+$ with $|v| = 2$. Now, $|v|_{x_n,\theta(x_n)} \geq 1$ and $|v|_{x_{n-1},\theta(x_{n-1})} \geq 1$. Then, by Lemma 5.5, $x_1 x_2 \cdots x_{n-2} x_{n-1} x_n \in C_\theta{}^{4n-11+4}(w)$, i.e., $x_1 x_2 \cdots x_{n-2} x_{n-1} x_n \in C_\theta{}^{4n-7}(w)$. Therefore, by Lemma 5.2, $x_1 x_2 \cdots x_{n-2} x_{n-1} x_n \in C_\theta{}^{4n-6}(w)$.

If $w = \alpha_1 \theta(x_1)\beta_1$, then $\theta(\beta_1)x_1\alpha_1 \in C_\theta(w)$ and $\theta(\alpha_1)\theta(\beta_1)x_1 \in C_\theta{}^2(w)$. Then, similar to the above, $x_1 x_2 \cdots x_{n-2} x_{n-1} x_n \in C_\theta{}^{2+4(n-3)+4}(w)$, i.e., $x_1 x_2 \cdots x_{n-2} x_{n-1} x_n \in C_\theta{}^{4n-6}(w)$.

Therefore, $x \in C_\theta^{4n-6}(w)$. Then, by Lemma 5.2, $x \in C_\theta^m(w)$ for all $m \geq 4n - 6$. Thus, $T \subseteq C_\theta^m(w)$ for all $m \geq 4n - 6$. Hence, we have the desired result. $\qquad\square$

One can observe by direct computation that if $w \in \Sigma$ and $w' \in \Sigma^2$, then $C_\theta(w) = C_\theta^*(w)$ and $C_\theta^3(w') = C_\theta^*(w')$. For words u of length at least three, we have the following result from Proposition 5.6.

Theorem 5.7. *If $w \in \Sigma^n$ with $n \geq 3$, then $C_\theta^{4n-6}(w) = C_\theta^*(w) = \{u \mid |u|_{a,\theta(a)} = |w|_{a,\theta(a)} \; \forall a \in \Sigma\}$.*

Proof: From the definition of iterated θ-conjugate of w, it is clear that $C_\theta^{4n-6}(w) \subseteq C_\theta^*(w)$. Consider $x \in C_\theta^*(w)$. Then, for some integer k, $x \in C_\theta^k(w)$. If $k < 4n - 6$, then by Lemma 5.2, $C_\theta^k(w) \subseteq C_\theta^{4n-6}(w)$. This implies $x \in C_\theta^{4n-6}(w)$. If $k \geq 4n - 6$, then by Proposition 5.6, $C_\theta^{4n-6}(w) = C_\theta^k(w) = \{u \mid |u|_{a,\theta(a)} = |w|_{a,\theta(a)} \; \forall a \in \Sigma\}$. This implies $x \in C_\theta^{4n-6}(w)$. Thus, $C_\theta^*(w) \subseteq C_\theta^{4n-6}(w)$. Hence, $C_\theta^{4n-6}(w) = C_\theta^*(w)$. $\qquad\square$

We now provide an example to show that for a word w of length $n \geq 3$, the bound $4n - 6$ is optimal. Additionally, we present an example to show that the smallest i such that $C_\theta^i(w) = C_\theta^*(w) = \{u \mid |u|_{a,\theta(a)} = |w|_{a,\theta(a)} \; \forall a \in \Sigma\}$ may or may not be equal to $4n - 6$.

Example 5.8. *Let $w \in \Sigma^*$ and θ be such that $\theta(a) = b$, $\theta(c) = d$ and $\theta(g) = h$.*

1. *Consider $w = cag$. Then,*
   $C_\theta(w) = \{hca, cag, hbd, hbc\}$,
   $C_\theta^2(w) = C_\theta(w) \cup \{bdg, bdh, cah, dhb, dah, bhc, chb, dag\}$,
   $C_\theta^3(w) = C_\theta^2(w) \cup \{hcb, gcb, gbc, gda, dbh, dgb, gca, dga, hda, ach, agc, gbd, agd, adh\}$,
   $C_\theta^4(w) = C_\theta^3(w) \cup \{bgd, ahc, cgb, gdb, gad, bgc, adg, bch, cha, bhd, dha, bcg, gac, ahd\}$,
   $C_\theta^5(w) = C_\theta^4(w) \cup \{cbh, cga, acg, had, cbg, dbg, hdb\}$,
   $C_\theta^6(w) = C_\theta^5(w) \cup \{hac\}$, *and $C_\theta^i(w) = C_\theta^6(w)$ for all $i \geq 6$.*
   Hence, $C_\theta^*(w) = C_\theta^6(w)$. Note that, for $m = 6 = 4|w| - 6$, $C_\theta^*(w) = C_\theta^m(w)$.

2. *Consider $u = caa$. Then, $C_\theta(w) = \{caa, bca, bbc, bbd\}$,*
   $C_\theta^2(w) = C_\theta(w) \cup \{bdb, cbb, cab, daa, dab, dbb, bda\}$,
   $C_\theta^3(w) = C_\theta^2(w) \cup \{bcb, abc, adb, acb, aad, ada, aca, abd, aac\}$,
   $C_\theta^4(w) = C_\theta^3(w) \cup \{bac, dba, bad, cba\}$, *and $C_\theta^i(w) = C_\theta^4(w)$ for all $i \geq 4$.*
   Hence, $C_\theta^*(w) = C_\theta^4(w)$. Note that, for $m = 4 < 4|u| - 6 = 6$, $C_\theta^*(w) = C_\theta^m(w)$.

For $w \in \Sigma^*$, $C_\theta^*(w) = \{u \mid |u|_{a,\theta(a)} = |w|_{a,\theta(a)} \; \forall a \in \Sigma\}$. Using basic counting principles, we now compute the number of elements in $C_\theta^*(w)$ for $w \in \Sigma^n$.

Theorem 5.9. *For $w \in \Sigma^n$, let $Alph(w) = A_1 \cup A_2 \cup B$ where $A_1 \cap A_2 = \emptyset$, $A_1 \cap B = \emptyset$, $A_2 \cap B = \emptyset$, $B = \{b : \theta(b) = b\}$, and $A_2 = \{\theta(a) : a \in A_1 \text{ and } \theta(a) \in Alph(w)\}$. Consider $A_1 = \{a_1, a_2, \cdots, a_l\}$ and $B = \{b_1, b_2, \cdots, b_r\}$ for some non-negative integers l and r. For each $a \in A_1$ and $b \in B$, let $|w|_{a,\theta(a)} = n_a$ and $|w|_{b,\theta(b)} = m_b$, respectively. Consider $\sum_{a \in A_1} n_a = n'$. Then*

$$|C_\theta^*(w)| = \binom{n}{n_{a_1} \; n_{a_2} \; \cdots \; n_{a_l} \; m_{b_1} \; m_{b_2} \; \cdots \; m_{b_r}} \times 2^{n'}.$$

Proof: Each word $u \in C_\theta{}^*(w)$ satisfies the condition $|u|_{c,\theta(c)} = |w|_{c,\theta(c)}$ for all $c \in \Sigma$. This means that what truly matters for counting the words in $C_\theta{}^*(w)$ is the total number of occurrences of each letter and its counterpart under θ.

Consider $Alph(w) = A_1 \cup A_2 \cup B$ where $A_1 \cap A_2 = \emptyset$, $A_1 \cap B = \emptyset$, $A_2 \cap B = \emptyset$, $B = \{b : \theta(b) = b\} = \{b_1, b_2, \cdots, b_r\}$, $A_1 = \{a_1, a_2, \cdots, a_l\}$ and $A_2 = \{\theta(a) : a \in A_1 \text{ and } \theta(a) \in Alph(w)\}$. Then A_1 is the collection of all those elements of $Alph(w)$ such that for any two distinct indices i and j with $1 \le i, j \le l$, we have $a_i \ne \theta(a_i)$, $a_j \ne \theta(a_j)$ and $a_i \ne \theta(a_j)$.

Let $C_1 = \{c_{a_1}, c_{a_2}, \cdots, c_{a_l}\}$ and $D = \{c_{b_1}, c_{b_2}, \cdots, c_{b_r}\}$ such that $|C_1| = |A_1| = l$, $|D| = |B| = r$, $C_1 \cap D = \emptyset$, $\Sigma \cap C_1 = \emptyset$ and $\Sigma \cap D = \emptyset$. To count the words of $C_\theta{}^*(w)$ systematically, we follow the following steps:

Step 1:

For each $a_i \in A_1$ ($1 \le i \le l$), we first replace each a_i and $\theta(a_i)$ in w by C_{a_i}. For each $b_j \in B$ ($1 \le j \le r$), we also replace each b_j in w by C_{b_j}. Then w becomes a word over the alphabet $C_1 \cup D$. Now by the given conditions, $|w|_{C_{a_i}} = n_{a_i}$ and $|w|_{C_{b_j}} = m_{b_j}$ for each $1 \le i \le l$ and $1 \le j \le r$.

Explanation of Step 1 through an Example: Consider $w = a_1 b_1 \theta(a_1) \theta(a_2)$ where $a_1, b_1, a_2 \in \Sigma$, $a_1 \ne \theta(a_1)$, $a_2 \ne \theta(a_2)$, and $b_1 = \theta(b_1)$. Then, $A_1 = \{a_1, \theta(a_2)\}$, $A_2 = \{\theta(a_1)\}$ and $B = \{b_1\}$. Let $C_1 = \{c_{a_1}, c_{\theta(a_2)}\}$ and $D = \{c_{b_1}\}$. Now, a_1 and $\theta(a_1)$ is replaced by C_{a_1}; $\theta(a_2)$ is replaced by $C_{\theta(a_2)}$; and b_1 is replaced by C_{b_1}. Then, w becomes $C_{a_1} C_{b_1} C_{a_1} C_{\theta(a_2)}$.

Step 2:

Now, we collect all the possible arrangements of w in a set S. Using multinomial coefficient, we have

$$|S| = \binom{n}{n_{a_1} \; n_{a_2} \; \cdots \; n_{a_l} \; m_{b_1} \; m_{b_2} \; \cdots \; m_{b_r}}.$$

Explanation of Step 2 through an Example: After Step 1, we have $w = C_{a_1} C_{b_1} C_{a_1} C_{\theta(a_2)}$. Then, $S = \{C_{a_1} C_{b_1} C_{a_1} C_{\theta(a_2)}, C_{a_1} C_{b_1} C_{\theta(a_2)} C_{a_1}, C_{a_1} C_{a_1} C_{b_1} C_{\theta(a_2)}, C_{a_1} C_{a_1} C_{\theta(a_2)} C_{b_1}, C_{a_1} C_{\theta(a_2)} C_{a_1} C_{b_1},$ $C_{a_1} C_{\theta(a_2)} C_{b_1} C_{a_1}, C_{b_1} C_{a_1} C_{a_1} C_{\theta(a_2)}, C_{b_1} C_{a_1} C_{\theta(a_2)} C_{a_1}, C_{b_1} C_{\theta(a_2)} C_{a_1} C_{a_1}, C_{\theta(a_2)} C_{a_1} C_{a_1} C_{b_1},$ $C_{\theta(a_2)} C_{a_1} C_{b_1} C_{a_1}, C_{\theta(a_2)} C_{b_1} C_{a_1} C_{a_1}\}$. Using multinomial coefficients, $|S| = \binom{4}{2 \; 1 \; 1} = 12$.

Step 3:

Consider $\alpha \in S$. Then $|\alpha|_{C_{a_i}} = n_{a_i}$ and $|\alpha|_{C_{b_j}} = m_{b_j}$ for each $1 \le i \le l$ and $1 \le j \le r$. We now replace each C_{a_i} in α by one of a_i and $\theta(a_i)$ where $1 \le i \le l$. Thus, each position of α with C_{a_i} has two possible replacements a_i and $\theta(a_i)$ where $1 \le i \le l$. Similarly, we replace each C_{b_j} in α by b_j where $1 \le j \le r$. Thus, each position of α with C_{b_j} has exactly one possible replacement b_j where $1 \le j \le r$. Then as $\sum_{i=1}^{l} n_{a_i} = n'$, we get $2^{n'}$ distinct words over Σ from α. Let's keep these $2^{n'}$ distinct words in a new set R. For each $\alpha \in S$, we get $2^{n'}$ distinct elements, and we put them in the set R. Then by basic counting principle,

$$|R| = \binom{n}{n_{a_1} \; n_{a_2} \; \cdots \; n_{a_l} \; m_{b_1} \; m_{b_2} \; \cdots \; m_{b_r}} \times 2^{n'}.$$

> Explanation of Step 3 through an Example: Consider $\alpha = C_{a_1} C_{a_1} C_{b_1} C_{\theta(a_2)}$. Then, C_{a_1} is replaced by one of a_1 and $\theta(a_1)$. $C_{\theta(a_2)}$ is replaced by one of $\theta(a_2)$ and a_2. C_{b_1} is replaced by b_1. Since $\sum_{a \in A_1} n_a = 3$, from α we got the following 2^3 elements: $a_1 a_1 b_1 \theta(a_2)$, $a_1 a_1 b_1 a_2$, $a_1 \theta(a_1) b_1 a_2$, $a_1 \theta(a_1) b_1 \theta(a_2)$, $\theta(a_1) a_1 b_1 a_2$, $\theta(a_1) a_1 b_1 \theta(a_2)$, $\theta(a_1)\theta(a_1) b_1 a_2$, $\theta(a_1)\theta(a_1) b_1 \theta(a_2)$. So, $|R| = 12 \times 8 = 96$.

Step 4:

We now show that $R = C_\theta^*(w)$. Consider $u \in R$. Then by the above construction, $|u|_{a,\theta(a)} = |w|_{a,\theta(a)}$ for all $a \in \Sigma$. Thus, $u \in C_\theta^*(w)$. Therefore, $R \subseteq C_\theta^*(w)$. Consider $v \in C_\theta^*(w)$. Then, $|v|_{a,\theta(a)} = |w|_{a,\theta(a)}$ for all $a \in \Sigma$. Now, we replace each a_i and $\theta(a_i)$ in v and w by C_{a_i} where $1 \le i \le l$. Also, we replace each b_j in v and w by C_{b_j} where $1 \le j \le r$. After replacements, consider v and w become v' and w', respectively, over $C_1 \cup D$. Since $|v|_{a,\theta(a)} = |w|_{a,\theta(a)}$ for all $a \in \Sigma$, v' is a permutation of w', i.e., $v' \in S$. Then by step 3, $v \in R$. Thus, $C_\theta^*(w) \subseteq R$. Therefore, $R = C_\theta^*(w)$. $\qquad\square$

We demonstrate Theorem 5.9 using the following example.

Example 5.10. *Let $w \in \Sigma^*$ and θ be such that $\theta(a) = b$, $\theta(c) = d$, $\theta(g) = h$ and $\theta(e) = e$.*

1. *Consider $w = cag$. Then $A_1 = \{a, c, g\}$, $A_2 = \emptyset$, $B = \emptyset$, $|w|_{a,b} = n_a = 1$, $|w|_{c,d} = n_c = 1$, $|w|_{g,h} = n_g = 1$, and $n_a + n_c + n_g = n' = 3$. Thus by Theorem 5.9, $|C_\theta^*(w)| = \binom{3}{1\ 1\ 1} \times 2^3 = 48$. In Example 5.8, we have explicitly computed the set $C_\theta^*(cag)$. From this set, we can verify that $|C_\theta^*(cag)| = 48$.*

2. *Consider $w = caa$. Then, $A_1 = \{a, c\}$, $A_2 = \emptyset$, $B = \emptyset$, $|w|_{c,d} = n_c = 1$, $|w|_{a,b} = n_a = 2$, and $n_c + n_a = n' = 3$. Thus by Theorem 5.9, $|C_\theta^*(w)| = \binom{3}{1\ 2} \times 2^3 = 24$. In Example 5.8, we have explicitly computed the set $C_\theta^*(caa)$. From this set, we can verify that $|C_\theta^*(caa)| = 24$.*

3. *Consider $w = aee$. Then, $A_1 = \{a\}$, $A_2 = \emptyset$, $B = \{e\}$, $|w|_{a,b} = n_a = 1$, $|w|_e = m_e = 2$, and $n_a = n' = 1$. Thus by Theorem 5.9, $|C_\theta^*(w)| = \binom{3}{1\ 2} \times 2^1 = 6$. We now verify the cardinality of $C_\theta^*(w)$ by computing it. Here, $C_\theta(w) = \{eeb, eea, eae, aee\}$, $C_\theta^2(w) = \{eeb, eea, bee, eae, ebe, aee\}$, and $C_\theta^i(w) = C_\theta^2(w)$ for all $i \ge 2$. Thus, $C_\theta^*(w) = C_\theta^2(w)$ and $|C_\theta^*(w)| = 6$.*

Consider L is a finite language and $\max\{|w| : w \in L\} \le 2$. Then by direct computation, $C_\theta^3(L) = C_\theta^*(L) = \bigcup_{w \in L} \{u \mid |u|_{a,\theta(a)} = |w|_{a,\theta(a)} \ \forall a \in \Sigma\}$. If the length of the largest word in L is greater than two, then by Theorem 5.7, we have the following remark.

Remark 5.11. *If L is a finite language and $n = \max\{|w| : w \in L\} \ge 3$, then $C_\theta^{4n-6}(L) = C_\theta^*(L) = \bigcup_{w \in L} \{u \mid |u|_{a,\theta(a)} = |w|_{a,\theta(a)} \ \forall a \in \Sigma\}$.*

It can be observed from Remark 5.11 that if L is finite, then $C_\theta^i(L)$ is finite for $i \ge 0$. Also, if $C_\theta^i(L)$ is finite for some $i \ge 0$, then as $L \subseteq C_\theta^i(L)$, L is finite. This implies that L is finite if and only if $C_\theta^i(L)$ is finite for $i \ge 0$.

We now study closure properties of families of regular and context-free languages under iterated θ-conjugate operation.

Lemma 5.12. *Regular languages and context-free languages are not closed under iterated θ-conjugate operation. Furthermore, the iterated θ-conjugate of a regular language is not necessarily context-free.*

Proof: Consider a regular language $L = \{(bce)^*\}$ over $\{b, c, d, e, f\}$ where $\theta(b) = b$, $\theta(c) = d$ and $\theta(e) = f$. Since $C_\theta^*(w) = \{u \mid |u|_{a,\theta(a)} = |w|_{a,\theta(a)} \ \forall a \in \Sigma\}$, $C_\theta^*(L) = \bigcup_{w \in L} \{u \mid |u|_{a,\theta(a)} = |w|_{a,\theta(a)} \ \forall a \in \Sigma\}$. Then, $C_\theta^*(L) \cap \{b^{m_1} c^{m_2} e^{m_3} \mid m_i \geq 0\} = \{b^n c^n e^n \mid n \geq 0\}$, which is not context-free (using Theorem 4.4). Note that $\{b^{m_1} c^{m_2} e^{m_3} \mid m_i \geq 0\}$ is a regular language. Since the regular languages and the context-free languages are closed under intersection with regular languages (Shallit (2008)), the claim holds.   □

6 θ-conjugate-free Languages

In this section, we first define a θ-conjugate-free language L, i.e., there are no two distinct strings in L such that one is a θ-conjugate of the other. Then for given $u, v \in \Sigma^n$, we discuss a linear time algorithm to decide whether or not v belongs to $C_\theta(u)$. Finally, we decide the existence of u, v where $u \neq v$ and $v \in C_\theta(u)$ in different language classes, i.e., we investigate some decidability problems for θ-conjugate-freeness for different language classes.

Definition 6.1. *A language L is θ-conjugate-free if for all $w \in L$, $C_\theta(w) \cap L = \{w\}$.*

6.1   *Decidability of θ-conjugate-freeness*

We first examine a straightforward scenario where we have two strings of equal length. Our task is to decide whether one string is a θ-conjugate of the other. Specifically, given two strings u and v, we need to decide whether or not v belongs to $C_\theta(u)$. A similar question for conjugates can be decided in linear time, i.e., given two strings u and v of equal length, we can decide whether $v \in C(u)$ in linear time. This can be done by checking whether v is a substring of uu.

For a word $w \in \Sigma^*$, let $w[i]$ and $w[i \cdots j]$ denote the letter of the string w at position i and the factor $w[i]w[i+1] \cdots w[j]$ of w, respectively, where $1 \leq i \leq j \leq |w|$. To check whether v belongs to $C_\theta(u)$, we first check the equality of u and v. If $u = v$, then $v \in C_\theta(u)$. Otherwise, we compute $\theta(u)$ and compare two strings, v and $\theta(u)$. Now, we find the first index where v and $\theta(u)$ have different characters. Let this index be i_L. If such an index does not exist, then $v = \theta(u)$ and $v \in C_\theta(u)$. If $i_L = 1$, then $v \notin C_\theta(u)$. If $i_L > 1$, then we check the equality of substrings $\theta(v[i_L \cdots |v|])$ and $\theta(u)[i_L \cdots |u|]$. If they are equal, then $v = \theta(y)x$ and $\theta(u) = \theta(y)\theta(x)$, i.e., $v \in C_\theta(u)$. Otherwise, $v \notin C_\theta(u)$. See the Algorithm 1 for details. From the analysis of Algorithm 1, we conclude that determining whether v is in $C_\theta(u)$ can be done in linear time.

Theorem 6.2. *If we have two strings u and v of equal length n, then we can decide whether or not v is a member of $C_\theta(u)$ in $O(n)$ time.*

We now investigate decidability problems for θ-conjugate-freeness of L when L is either regular or context-free. Endrullis et al. (2017) proved that it is decidable to determine whether or not a regular language contains two distinct conjugates of a word. We now prove that it is also decidable to determine whether or not a regular language L with no θ-palindrome contains two distinct θ-conjugates of a word.

Theorem 6.3. *If L is a regular language with no θ-palindrome, then it is decidable to determine whether or not L is θ-conjugate-free.*

Proof: Consider L is a regular language with no θ-palindromes. From Proposition 3.4, we know that $C_\theta(u'a) = \{u'a\} \cup \theta(a)C_\theta(u')$ for $u' \in \Sigma^*$ and $a \in \Sigma$. Then, $C_\theta(L) = \bigcup_{w \in L} C_\theta(w) = L \cup$

Algorithm 1 A linear time algorithm for deciding $v \in C_\theta(u)$

Require: Two strings of u and v of same length n and mapping θ

Define an empty array A of size n

$z \leftarrow \lambda$

$s \leftarrow 1$

$k \leftarrow 0$

$k_1 \leftarrow 0$

for $j = 1$ to n **do**

    if $u[j] = v[j]$ **then** $k \leftarrow k + 1$

    end if

end for

if $k = n$ **then** return true $\triangleright$ Then, $v = u$

else

    for $l = n$ to 1 **do** $z \leftarrow z \,||\, \theta(u[l])$ $\triangleright \,||$ indicates concatenation operation

    end for

end if

$\theta(u) \leftarrow z$

if $v[1] \neq \theta(u)[1]$ **then** return false

else

    for $i = 2$ to n **do**

        if $\theta(u)[i] \neq v[i]$ **then** $A[s] \leftarrow i; s \leftarrow s + 1$

            for $l = i$ to n **do**

                if $\theta(u)[i] \neq \theta(v[n + A[1] - i])$ **then** return false

                end if

            end for

            return true $\triangleright \, \theta(u) = \theta(y)\theta(x)$, $v = \theta(y)x$ for some $x, y \in \Sigma^+$

        else

            $k_1 \leftarrow k_1 + 1$

        end if

    end for

    if $k_1 = n - 1$ **then** return true $\triangleright$ Then, $v = \theta(u)$

    end if

end if

$\bigcup\limits_{\substack{u'a \in L \\ u' \in \Sigma^*, a \in \Sigma}} \theta(a)C_\theta(u') = L \cup T$ where $T = \bigcup\limits_{\substack{u'a \in L \\ u' \in \Sigma^*, a \in \Sigma}} \theta(a)C_\theta(u')$. Now, using a similar proof tech-

nique of Theorem 4.3, it is not hard to prove that if L is regular, then T is also regular. We now prove that L is θ-conjugate-free if and only if $L \cap T = \emptyset$. Consider L is θ-conjugate-free. Then for all $w \in L$, $C_\theta(w) \cap L = \{w\}$. If $x \in L \cap T$, then $x \in L$ and $x \in T$. This implies $x \in \theta(a)C_\theta(z)$ for some $za \in L$. If $x = za$, then for $\alpha \in \Sigma^+$ and $\beta \in \Sigma^*$, $x = \theta(\alpha)\beta$, $za = \beta\alpha$ and $\theta(\alpha)\beta = \beta\alpha$ which implies x is a θ-palindrome, a contradiction to the fact that L does not contain any θ-palindrome. Thus, $x \neq za$, i.e., $\{x, za\} \subseteq C_\theta(za) \cap L$ which implies L is not θ-conjugate-free, a contradiction. Hence, $L \cap T = \emptyset$. Now, consider $L \cap T = \emptyset$. If $w, w' \in L$ such that $w \neq w'$ and $w' \in C_\theta(w)$, then $w' \in T$ which implies $w' \in L \cap T$, a contradiction. Thus, L is θ-conjugate-free. Therefore, L is θ-conjugate-free if and only if $L \cap T = \emptyset$. Now, L and T are regular languages. Since the emptiness of the intersection of two regular languages is decidable (Shallit (2008)), we can decide whether or not L is θ-conjugate-free. $\square$

Now, we discuss the decidability result for a context-free language. First, we recall the following. An instance of the Post's Correspondence Problem (PCP) Post (1946) consists of a positive integer n, and two ordered n-tuples of strings (U, V), where $U = (u_0, u_1, \ldots, u_{n-1})$ and $V = (v_0, v_1, \ldots, v_{n-1})$, $u_i, v_i \in \Sigma^*, 0 \le i \le n-1$. A solution for the PCP instance (U, V) is a sequence of integers $i_1, i_2, \ldots, i_k$, $0 \le i_j \le n-1, j = 1, 2, \ldots, k, k \ge 1$, such that

$$u_{i_1}u_{i_2} \cdots u_{i_k} = v_{i_1}v_{i_2} \cdots v_{i_k}.$$

Proposition 6.4. *Post (1946) The decision problem of determining whether or not a given PCP instance has a solution is undecidable.*

Using PCP, we can show that it is undecidable to determine whether or not a context-free language contains two distinct conjugates of a word (the proof is provided in the Appendix). We now prove that determining the θ-conjugate-freeness of a given context-free language is also undecidable by reducing the PCP to this problem.

Theorem 6.5. *It is undecidable to determine whether or not a given context-free language L is θ-conjugate-free.*

Proof: Let Σ be alphabet and (U, V) be an instance of PCP problem where $U = (u_0, u_1, \ldots, u_{n-1})$, $V = (v_0, v_1, \ldots, v_{n-1})$ with each $u_i, v_i \in \Sigma^*$. Assume that the symbols $\#, \$, a_0, a_1, \cdots, a_{n-1}$ are distinct and not in Σ. Consider $\Sigma' = \Sigma \cup \{\#, \$, a_0, a_1, \cdots, a_{n-1}\}$. Let $\theta : \Sigma'^* \to \Sigma'^*$ be an antimorphic involution defined as

1. $\theta(a) \in \Sigma$ for all $a \in \Sigma$

2. $\theta(\$) = \#$

3. $\theta(c) = c$ for all $c \in \{a_0, a_1, \cdots, a_{n-1}\}$.

We define a context-free grammar $G = (N, \Sigma', P, S)$ where $N = \{S, N_U, N_V\}$ is the set of non-terminals, Σ' is the set of terminals, S is the start symbol and P is the set of derivation rules which contains the following rules:

- $S \to \#N_U\# \mid \$N_V\$$

- $N_U \to u_0 N_U a_0 \mid u_1 N_U a_1 \mid \cdots \mid u_{n-1} N_U a_{n-1} \mid \lambda$

- $N_V \to a_0 N_V \theta(v_0) \mid a_1 N_V \theta(v_1) \mid \cdots \mid a_{n-1} N_V \theta(v_{n-1}) \mid \lambda$

Now, $L(G)$ denotes the set of all words generated by the grammar G and any string of $L(G)$ is in one of the following forms :

$$\# u_{j_1} u_{j_2} \cdots u_{j_l} a_{j_l} \cdots a_{j_2} a_{j_1} \# \tag{A}$$

or

$$\$ a_{j_1} a_{j_2} \cdots a_{j_l} \theta(v_{j_l}) \cdots \theta(v_{j_2}) \theta(v_{j_1}) \$ \tag{B}$$

Under our assumed conditions on $\#, \$, a_0, a_1, \cdots, a_{n-1}, \theta, u_0, u_1, \cdots u_{n-1}, v_0, v_1, \cdots, v_{n-1}$, we observe that the symbol $\#$ appears only at the left and right ends of words in the form A, the symbol $\$$ does not appears in words of the form A, the symbol $\$$ appears only at the left and right ends of words in the form B, and the symbol $\#$ does not appears in words of the form B. We now show that $L(G)$ is not θ-conjugate-free if and only if the PCP instance (U, V) has a solution.

First, we prove that if the PCP instance (U, V) has a solution, then $L(G)$ is not θ-conjugate-free. Assume that the PCP instance (U, V) has a solution. Then there is a sequence of integers $i_0, i_1, \ldots, i_k$ such that $u_{i_0} u_{i_1} \cdots u_{i_k} = v_{i_0} v_{i_1} \cdots v_{i_k}$. Now, $L(G)$ contains the following elements:

$$\begin{aligned} \beta' = \$ a_{i_0} a_{i_1} \cdots a_{i_k} \theta(v_{i_k}) \cdots \theta(v_{i_1}) \theta(v_{i_0}) \$ &= \$ a_{i_0} a_{i_1} \cdots a_{i_k} \theta(v_{i_0} v_{i_1} \cdots v_{i_k}) \$ \\ &= \$ a_{i_0} a_{i_1} \cdots a_{i_k} \theta(u_{i_0} u_{i_1} \cdots u_{i_k}) \$ \end{aligned}$$

and

$$\alpha' = \# u_{i_0} u_{i_1} \cdots u_{i_k} a_{i_k} \cdots a_{i_1} a_{i_0} \#.$$

As $\theta(\$) = \#$ and $\theta(c) = c \; \forall c \in \{a_0, a_1, \cdots, a_{n-1}\}$, we have $\beta' = \theta(\alpha')$. Since $\# \neq \$$, $\alpha' \neq \beta'$. Thus, $L(G)$ is not θ-conjugate-free.

Now, we prove the other part, i.e., if $L(G)$ is not θ-conjugate-free, then the PCP instance (U, V) has a solution. Assume that $L(G)$ is not θ-conjugate-free, i.e., there exist w_1, w_2 in $L(G)$ such that $w_2 \in C_\theta(w_1)$ and $w_1 \neq w_2$. Then $w_1 = xy$ and $w_2 = \theta(y)x$ for some $x \in \Sigma'^*$ and $y \in \Sigma'^+$. Since elements of $L(G)$ are in the form A or B, we have two cases:

- **Case I :** Consider $w_1 = xy$ is in the form A. Then for some integer m,

$$w_1 = xy = \# u_{i_1} u_{i_2} \cdots u_{i_m} a_{i_m} \cdots a_{i_2} a_{i_1} \#.$$

As $\theta(\#) = \$$ and $y \in \Sigma'^+$, $\theta(y)x$ starts with $\$$. This implies $\theta(y)x$ must be in the form B. Then for some integer r,

$$w_2 = \theta(y)x = \$ a_{j_1} a_{j_2} \cdots a_{j_r} \theta(v_{j_r}) \cdots \theta(v_{j_2}) \theta(v_{j_1}) \$.$$

Since $|w_1|_\$ = 0$, $\theta(\#) = \$$ and $w_2 \in C_\theta(w_1)$, we have $w_2 = \theta(w_1)$. Then using the values of w_2 and w_1, we have

$$\begin{aligned} \$ a_{j_1} a_{j_2} \cdots a_{j_r} \theta(v_{j_r}) \cdots \theta(v_{j_2}) \theta(v_{j_1}) \$ &= \theta(\# u_{i_1} u_{i_2} \cdots u_{i_m} a_{i_m} \cdots a_{i_2} a_{i_1} \#) \\ \implies \$ a_{j_1} a_{j_2} \cdots a_{j_r} \theta(v_{j_r}) \cdots \theta(v_{j_2}) \theta(v_{j_1}) \$ &= \$ a_{i_1} a_{i_2} \cdots a_{i_m} \theta(u_{i_m}) \cdots \theta(u_{i_2}) \theta(u_{i_1}) \$. \end{aligned}$$

Since $\{a_0, a_1, \cdots, a_{n-1}\} \cap \Sigma = \emptyset$, each $u_i, v_i \in \Sigma^*$, and $\theta(a) \in \Sigma \; \forall a \in \Sigma$, we have, $r = m$, $i_1 = j_1, i_2 = j_2, \ldots, i_m = j_m$ and $v_{i_1} v_{i_2} \cdots v_{i_m} = u_{i_1} u_{i_2} \cdots u_{i_m}$. Hence, the PCP instance has a solution.

- **Case II :** Consider $w_1 = xy$ is in the form B. Then similar to Case I, we show that the PCP instance has a solution.

Therefore, as PCP is undecidable Post (1946), determining whether or not L is θ-conjugate-free is also undecidable when L is context-free. $\qquad\square$

7 Conclusions

This work explores various properties of the set of all θ-conjugates of a word and language. First, we have discussed the structure of θ-conjugates of a word and the solution of $C_\theta(u) = C_\theta(v)$. Then we have investigated closure properties of certain families of languages under θ-conjugate operation. Next, we have studied the iterated θ-conjugate of words and languages. Finally, we have discussed the concept of θ-conjugate-free language and some related decidability problems. For regular language L with no θ-palindrome, it is decidable to determine whether or not L is θ-conjugate-free. But for an arbitrary regular language L, the answer of the following question is still unknown.

Question 1: For a regular language L, is it decidable to determine whether or not L is θ-conjugate-free?

Acknowledgements

References

L. M. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266(5187): 1021–1024, 1994.

Y. Aoi, T. Yoshinobu, K. Tanizawa, K. Kinoshita, and H. Iwasaki. Solution of the knapsack problem by deoxyribonucleic acid computing. *Japanese Journal of Applied Physics*, 37(10R):5839–5841, 1998.

Y. Belkheyar, J. Daemen, C. Dobraunig, S. Ghosh, and S. Rasoolzadeh. Introducing two low-latency cipher families: Sonic and supersonic. *Cryptology ePrint Archive*, 2023.

S. Bi, S. Yue, Q. Wu, and J. Ye. Initiator-catalyzed self-assembly of duplex-looped DNA hairpin motif based on strand displacement reaction for logic operations and amplified biosensing. *Biosensors and Bioelectronics*, 83:281–286, 2016.

R. S. Braich, N. Chelyapov, C. Johnson, P. W. Rothemund, and L. Adleman. Solution of a 20-variable 3-SAT problem on a DNA computer. *Science*, 296(5567):499–502, 2002.

A. Brandstädt. Closure properties of certain families of formal languages with respect to a generalization of cyclic closure. *RAIRO. Informatique Théorique*, 15(3):233–252, 1981.

E. Chiniforooshan, L. Kari, and Z. Xu. Pseudo-power avoidance. In *Developments in Language Theory*, volume 6224 of *Lecture Notes in Computer Science*, pages 432–433. Springer, 2010.

E. Czeizler, L. Kari, and S. Seki. On a special class of primitive words. *Theoretical Computer Science*, 411(3):617–630, 2010.

J. Endrullis, J. Shallit, and T. Smith. Undecidability and finite automata. In *Developments in Language Theory*, volume 10396 of *Lecture Notes in Computer Science*, pages 160–172. Springer, 2017.

P. Gawrychowski, F. Manea, and D. Nowotka. Testing generalised freeness of words. In *31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014)*, volume 25 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 337–349, 2014.

T. Head, G. Rozenberg, R. S. Bladergroen, C. Breek, P. Lommerse, and H. P. Spaink. Computing with DNA by operating on plasmids. *Biosystems*, 57(2):87–93, 2000.

C. V. Henkel, R. S. Bladergroen, C. I. Balog, A. M. Deelder, T. Head, G. Rozenberg, and H. P. Spaink. Protein output for DNA computing. *Natural Computing*, 4:1–10, 2005.

C. V. Henkel, T. Bäck, J. N. Kok, G. Rozenberg, and H. P. Spaink. DNA computing of solutions to knapsack problems. *Biosystems*, 88(1-2):156–162, 2007.

I. Jeddi and L. Saiz. Three-dimensional modeling of single stranded DNA hairpins for aptamer-based biosensors. *Scientific Reports*, 7(1):1178, 2017.

G. Jirásková and A. Okhotin. State complexity of cyclic shift. *RAIRO-Theoretical Informatics and Applications*, 42(2):335–360, 2008.

C. R. Johnson. Automating the DNA computer: solving n-variable 3-SAT problems. *Natural Computing*, 7:239–253, 2008.

L. Kari and M. S. Kulkarni. Generating the pseudo-powers of a word. *Journal of Automata, Languages and Combinatorics*, 19(1-4):157–171, 2014a.

L. Kari and M. S. Kulkarni. Pseudo-identities and bordered words. In *Discrete Mathematics and Computer Science*, pages 207–222. Editura Academiei Romãne, Romania, 2014b.

L. Kari and M. S. Kulkarni. Disjunctivity and other properties of sets of pseudo-bordered words. *Acta Informatica*, 54(4):379–398, 2017.

L. Kari and K. Mahalingam. Involutively bordered words. *International Journal of Foundations of Computer Science*, 18(05):1089–1106, 2007.

L. Kari and K. Mahalingam. Watson-crick bordered words and their syntactic monoid. *International Journal of Foundations of Computer Science*, 19(05):1163–1179, 2008a.

L. Kari and K. Mahalingam. Watson-Crick conjugate and commutative words. In *DNA Computing*, volume 4848 of *Lecture Notes in Computer Science*, pages 273–283. Springer, 2008b.

L. Kari and K. Mahalingam. Watson-Crick palindromes in DNA computing. *Natural Computing*, 9(2): 297–316, 2010.

L. Kari and T. Ng. State complexity of pseudocatenation. In *Language and Automata Theory and Applications*, volume 11417 of *Lecture Notes in Computer Science*, pages 203–214. Springer, 2019.

L. Kari and S. Seki. On pseudoknot-bordered words and their properties. *Journal of Computer and System Sciences*, 75(2):113–121, 2009.

P. Linz. *An introduction to formal languages and automata (5th edition)*. Jones & Bartlett Learning, 2012.

M. Lothaire. *Combinatorics on words*. Cambridge University Press, 1997.

R. C. Lyndon and M. P. Schützenberger. The equation $a^M = b^N c^P$ in a free group. *Michigan Mathematical Journal*, 9:289–298, 1962.

K. Mahalingam, A. Maity, and P. Pandoh. Counting (Watson-Crick) palindromes in Watson-Crick conjugates. *Information and Computation*, 285:104863, 2022.

T. Oshiba. Closure property of the family of context-free languages under the cyclic shift operation. *Electronics & Communications in Japan*, 55(4):119–122, 1972.

Q. Ouyang, P. D. Kaplan, S. Liu, and A. Libchaber. DNA solution of the maximal clique problem. *Science*, 278(5337):446–449, 1997.

E. L. Post. A variant of a recursively unsolvable problem. *Bulletin of the American Mathematical Society*, 52(4):264–268, 1946.

G. Rozenberg and A. Salomaa (Eds.). *Handbook of Formal Languages*, volume 1. Springer, 1997.

K. Sakamoto, H. Gouzu, K. Komiya, D. Kiga, S. Yokoyama, T. Yokomori, and M. Hagiya. Molecular computation by DNA hairpin formation. *Science*, 288(5469):1223–1226, 2000.

J. Shallit. *A second course in formal languages and automata theory*. Cambridge University Press, 2008.

B. Shi, Y. Zhou, Y. Huang, J. Zhang, and X. Li. Recent advances on the encoding and selection methods of DNA-encoded chemical library. *Bioorganic & Medicinal Chemistry Letters*, 27(3):361–369, 2017.

Š. Starosta. On theta-palindromic richness. *Theoretical Computer Science*, 412(12-14):1111–1121, 2011.

V. Walter, F. A. Wright, and A. B. Nobel. Consistent testing for recurrent genomic aberrations. *Biometrika*, 102(4):783–796, 2015.

X. Wang, Z. Bao, J. Hu, S. Wang, and A. Zhan. Solving the SAT problem using a DNA computing algorithm based on ligase chain reaction. *Biosystems*, 91(1):117–125, 2008.

D. Y. Zhang and G. Seelig. Dynamic DNA nanotechnology using strand-displacement reactions. *Nature Chemistry*, 3(2):103–113, 2011.

8 Appendix

A0. For $u, v \in \Sigma^*$ with $u \neq v$, consider $C_\theta(u) = C_\theta(v)$ and $|C_\theta(u)| \geq 3$. Then $u = a^{n_1} u' a^{n_2}$ and $v = u^R$ where $a \in \Sigma$, $u' \in \Sigma^+$, $a \notin Pref(u')$, $a \notin Suf(u')$, $u' = u'^R$, $n_1, n_2 \geq 1$, $n_1 \neq n_2$, and $\theta(c) = c$ for all $c \in Alph(u)$.

Proof: Let $u = u_1 u_2 \cdots u_n$ and $v = v_1 v_2 \cdots v_n$ be words of length n such that $u_i, v_i \in \Sigma$.

If $\theta(u_n) \neq u_1$, then in $C_\theta(u)$, u is the only element having u_1 as a prefix and the rest of the elements have $\theta(u_n)$ as a prefix. If $\theta(v_n) = v_1$, then all elements of $C_\theta(v)$ have the same letter as a prefix, which is a contradiction to the fact that $C_\theta(u) = C_\theta(v)$. This implies, $\theta(v_n) \neq v_1$. Then, v is the only element in $C_\theta(v)$ that starts with v_1. Also, u is the only element in $C_\theta(u)$ that starts with u_1. As $|C_\theta(u)| \geq 3$ and $C_\theta(u) = C_\theta(v)$, we get, $u = v$, which is a contradiction. Thus, $\theta(u_n) = u_1$. Then all elements in $C_\theta(u)$ start with u_1. Since, $C_\theta(u) = C_\theta(v)$, we get, $u_1 = v_1$ and $\theta(u_n) = u_1 = \theta(v_n)$, i.e., $u_n = v_n$. Let

$$u = u_1^{n_1} u' u_n^{n_2} \text{ and } v = u_1^{m_1} v' u_n^{m_2} \tag{1}$$

where $u', v' \in \Sigma^*$, $u_1 \notin Pref(u')$, $u_1 \notin Pref(v')$, $u_n \notin Suf(u')$, $u_n \notin Suf(v')$, and $n_1, n_2, m_1, m_2 \geq 1$.

Without loss of generality, let $n_1 + n_2 \geq m_1 + m_2$. Since $u = u_1^{n_1} u' u_n^{n_2}$, for an element in $C_\theta(u)$ to start with $u_1^{n_1 + n_2}$, it must be of the form $\theta(y u_n^{n_2}) u_1^{n_1} x$ where $xy = u'$. From Eq. (1), it is clear that u_n is not present as a suffix of y and u_1 is not present as a prefix of x. This implies that $u_1^{n_1 + n_2} u'$ is the only element in $C_\theta(u)$ having $u_1^{n_1 + n_2}$ as a prefix. Similarly, $u_1^{m_1 + m_2} v'$ is the only element in $C_\theta(v)$ having $u_1^{m_1 + m_2}$ as a prefix. If $n_1 + n_2 > m_1 + m_2$, then $u_1^{n_1 + n_2} u' \notin C_\theta(v)$, which is a contradiction to the fact that $C_\theta(u) = C_\theta(v)$. This implies $n_1 + n_2 = m_1 + m_2$. Also, we get, $u_1^{n_1 + n_2} u' = u_1^{m_1 + m_2} v'$, i.e., $u' = v'$. Thus, we get,

$$u = u_1^{n_1} u' u_n^{n_2} \text{ and } v = u_1^{m_1} u' u_n^{m_2}$$

and $n_1 + n_2 = m_1 + m_2$. If $n_1 = m_1$, then $n_2 = m_2$. This implies $u=v$, which is a contradiction. Thus, $n_1 \neq m_1$. Without loss of generality, let $n_1 > m_1$ and $m_2 > n_2$. We have the following cases:

- If $u' = \lambda$, then $u = u_1^{n_1} u_n^{n_2}$ and $v = u_1^{m_1} u_n^{m_2}$. If $u_1 = u_n$, then $u = v$, a contradiction. If $u_1 \neq u_n$, then as $n_1 > m_1 \geq 1$, we have, $n_1 \geq 2$ and $u_1^{n_2} u_n u_1^{n_1 - 1} \in C_\theta(u) \setminus C_\theta(v)$, which is a contradiction.

- If $u' \in \Sigma^+$, then $u = u_1^{n_1} u' u_n^{n_2}$ and $v = u_1^{m_1} u' u_n^{m_2}$. We have the following cases:

    - $n_1 \neq m_2$: If $u_1 \neq u_n$, then as $n_1 > m_1 \geq 1$, we have, $n_1 \geq 2$ and $u_1^{n_2} \theta(u') u_n u_1^{n_1 - 1} \in C_\theta(u) \setminus C_\theta(v)$, which is a contradiction. Otherwise, $u_1 = u_n$. Then $u = u_1^{n_1} u' u_1^{n_2}$ and $v = u_1^{m_1} u' u_1^{m_2}$. If $n_1 > m_2$, then $m_1 > n_2$ and $u_1^{n_2} \theta(u') u_1^{n_1} \in C_\theta(u) \setminus C_\theta(v)$, which is a contradiction. If $n_1 < m_2$, then $n_2 > m_1$, $u = u_1^{n_1} u' u_1^{m_1 + k}$ and $v = u_1^{m_1} u' u_1^{n_1 + k}$ for $k \geq 1$. We have the following cases:

        * $\theta(u') \neq u'$: Then, $u_1^{n_1 + k} \theta(u') u_1^{m_1} \in C_\theta(v) \setminus C_\theta(u)$, which is a contradiction.
        * $\theta(u') = u'$: Then, as $n_1 > m_1$, $v \in C_\theta(v) \setminus C_\theta(u)$, which is a contradiction.

    - $n_1 = m_2$: Then, $n_2 = m_1$. If $u_1 \neq u_n$, then since $n_1 > n_2 > 0$, we get, $n_1 \geq 2$ which implies $u_1^{n_2} \theta(u') \theta(u_1) u_1^{n_1 - 1} \in C_\theta(u) \setminus C_\theta(v)$, which is a contradiction. Thus, $u_1 = u_n$ and we get, $u = u_1^{n_1} u' u_1^{n_2}$ and $v = u_1^{n_2} u' u_1^{n_1}$. If $\theta(u') \neq u'$, then $u_1^{n_2} \theta(u') u_1^{n_1} \in C_\theta(u) \setminus C_\theta(v)$, which is a contradiction. Hence, $\theta(u') = u'$. We have the following:

* If $\theta(a) = a$ for all $a \in Alph(u)$, we have, $\theta(u') = u'^R$. Thus, $u' = u'^R$. This implies as $u = u_1^{n_1} u' u_1^{n_2}$ and $v = u_1^{n_2} u' u_1^{n_1}$, $u = v^R$.

* If there exists some $a \in Alph(u)$ such that $\theta(a) \neq a$, then as $\theta(u_1) = u_1$, we have, $u_1 \neq a$. Now, $\theta(u') = u'$ implies $u' = xbu''\theta(b)\theta(x)$ for $\theta(b) \neq b \in \Sigma$ and $x, u'' \in \Sigma^*$ such that $\theta(d) = d$ for all $d \in Alph(x)$. Since $\theta(u_1) = u_1$, we have, $u_1 \neq b$. Now, $u = u_1^{n_1} xbu''\theta(b)\theta(x)u_1^{n_2}$ and $v = u_1^{n_2} xbu''\theta(b)\theta(x)u_1^{n_1}$. Consider $w = u_1^{n_2} xb\theta(u'')u_1^{n_1} xb$. Then, $w \in C_\theta(u) \setminus C_\theta(v)$, which is a contradiction.

Hence, $u = u_1^{n_1} u' u_1^{n_2}$ and $v = u^R$ where $u_1 \in \Sigma$, $u' \in \Sigma^+$, $u_1 \notin Pref(u')$, $u_1 \notin Suf(u')$, $u' = u'^R$, $n_1, n_2 \geq 1$, $n_1 \neq n_2$, and $\theta(c) = c$ for all $c \in Alph(u)$. $\qquad\square$

A1. Prove that the language $L_1 = \{d^n c^k a^n b^k \mid n, k \geq 1\}$ is not context-free by pumping lemma.
Proof: Suppose $L_1 = \{d^n c^k a^n b^k \mid n, k \geq 1\}$ is context-free. Then, by pumping lemma (Theorem 4.4),

there exists a pumping length $p > 0$ such that any string $s \in L_1$ with $|s| \geq p$ can be written as $s = uvwxy$, satisfying the following: $|vwx| \leq p$, $|vx| \geq 1$ and $uv^i wx^i y \in L_1$ for all $i \geq 0$.

Let $s = d^p c^p a^p b^p$. Clearly, $s \in L_1$ and $|s| > p$. Then pumping lemma guarantees that s can be split into $uvwxy$ where $|vwx| \leq p$, $|vx| \geq 1$ and $uv^i wx^i y \in L_1$ for all $i \geq 0$. Since $|vwx| \leq p$, the factor vwx must be entirely contained with in one of d^p, c^p, a^p, b^p, $d^p c^p$, $c^p a^p$ or $a^p b^p$.

If vwx lies entirely in d^p, then pumping v and x changes only the number of d's in pumped strings. Then there exists some pumped strings that do not lie in L_1. So, vwx does not lie entirely in d^p. Similarly, we can show that vwx does not lie entirely in any of c^p, a^p and b^p. If vwx lies entirely in $d^p c^p$, then pumping v and x changes only the number of d's and c's in pumped strings. Then there exists some pumped strings that do not lie in L_1. So, vwx does not lie entirely in $d^p c^p$. Similarly, we can show that vwx does not lie entirely in any of $c^p a^p$ and $a^p b^p$. Therefore, in all possible cases, pumping v and x results in some strings that do not lie in L_1, which contradicts the pumping lemma. Thus, our assumption that L_1 is context-free is false. $\qquad\square$

A2. *The proof of Theorem 4.8:*

Proof: Consider L is a recursively enumerable language. Since the recursively enumerable languages are closed under reversal Rozenberg and Salomaa (Eds.), L^R is also recursively enumerable. Let $\psi : \Sigma^* \to \Sigma^*$ be a morphic involution such that $\psi(a) = \theta(a)$ for all $a \in \Sigma$. Consider

$$L' = \bigcup_{uv \in L^R} \{\psi(u)v^R\}.$$

Then, by Lemma 4.2,

$$L' = \bigcup_{uv \in L^R} \{\psi(u)v^R\} = \bigcup_{v^R u^R \in L} \{\theta(u^R)v^R\} = C_\theta(L).$$

We prove that L' is a recursively enumerable language. Since L^R is recursively enumerable, there exists an unrestricted grammar $G = (V, \Sigma, P, S)$ such that $L^R = L(G)$. We now construct an unrestricted grammar G_N such that $L(G_N) = L'$. Let $V_1 = \{S', T_1, T_2, T_3, T_4, T\}$ and $V_2 = \{X_a{}^o, X_a{}^I, X_a{}^D, X_a{}^R, X_a{}^T, X_a{}^F, X_a{}^N : a \in \Sigma\}$. Consider $G_N = (V_N, \Sigma, P_N, S')$ where $V_N = V \cup V_1 \cup V_2$, $V \cap V_1 = \emptyset$, $V \cap V_2 = \emptyset$, $V_1 \cap V_2 = \emptyset$, S' is the start symbol and P_N is the union of the following sets :

1. $P_N(1) = \{S' \to T_1 S T_2 \mid T_1 S T\}$

2. $P_N(2) = \{h(\alpha) \to h(\beta) \ : \ \alpha \to \beta \in P\}$

3. $P_N(3) = \{T_1 X_a^o \to X_a^N T_1 \ : \ a \in \Sigma\}$

4. $P_N(4) = \{X_a^o T_2 \to T_3 X_a^F \ : \ a \in \Sigma\}$

5. $P_N(5) = \{X_a^o T_3 \to T_3 X_a^o \ : \ a \in \Sigma\}$

6. $P_N(6) = \{X_a^o T \to T_4 X_a^o \ : \ a \in \Sigma\}$

7. $P_N(7) = \{X_a^N Z \to Z\theta(a) \ : a \in \Sigma\}$

8. $P_N(8) = \{X_a^N Z' \to Z'\theta(a) \ : \ a \in \Sigma\}$

9. $P_N(9) = \{Z \to \lambda, Z' \to \lambda, Z'' \to \lambda\}$

10. $P_N(10) = \{Z'' X_a^o \to X_a^D Z \ : \ a \in \Sigma\}$

11. $P_N(11) = \{X_a^D Z \to Za \ : \ a \in \Sigma\}$

12. $P_N(12) = \{Z X_a^o \to Z X_a^I \ : \ a \in \Sigma\}$

13. $P_N(13) = \{Z X_{a_1}^I X_{a_2}^o \to X_{a_1}^I Z X_{a_2}^R \ : \ a_1, a_2 \in \Sigma\}$

14. $P_N(14) = \{Z X_{a_1}^R X_{a_2}^o \to X_{a_1}^R Z X_{a_2}^R \ : \ a_1, a_2 \in \Sigma\}$

15. $P_N(15) = \{Z X_{a_1}^R X_{a_2}^F \to X_{a_2}^T Z X_{a_1}^F \ : \ a_1, a_2 \in \Sigma\}$

16. $P_N(16) = \{X_{a_1}^R X_{a_2}^T Z \to X_{a_2}^T Z X_{a_1}^o \ : \ a_1, a_2 \in \Sigma\}$

17. $P_N(17) = \{X_{a_1}^I X_{a_2}^T Z \to X_{a_2}^D Z X_{a_1}^I \ : \ a_1, a_2 \in \Sigma\}$

18. $P_N(18) = \{Z X_{a_1}^I X_{a_2}^F \to X_{a_2}^D X_{a_1}^D Z \ : \ a_1, a_2 \in \Sigma\}$

where $Z = T_1 T_3$, $Z' = T_1 T_2$, $Z'' = T_1 T_4$ and $h : (V \cup \Sigma)^* \to V_N$ is the morphism defined by $h(\lambda) = \lambda, h(B) = B$ for all $B \in V$ and $h(a) = X_a^o$ for all $a \in \Sigma$. We prove the following two claims:

Claim 1 : $L' \subseteq L(G_N)$: Consider $x \in L'$. Then, $x = \psi(y_1 y_2 \cdots y_i)(y_n y_{n-1} \cdots y_{i+1})$ where $y_1 y_2 \cdots y_n \in L^R$ and $y_i \in \Sigma$. This implies as $\psi(a) = \theta(a)$ for all $a \in \Sigma$, $x = \theta(y_1)\theta(y_2)\cdots\theta(y_i)(y_n y_{n-1} \cdots y_{i+1})$. Now, we show that $x \in L(G_N)$. Since $y_1 y_2 \cdots y_n = y \in L^R$, S derives y. Then, by $P_N(1)$, $P_N(2)$ and definition of h, S' derives $T_1 X_{y_1}^o X_{y_2}^o \cdots X_{y_n}^o T_2$ or $T_1 X_{y_1}^o X_{y_2}^o \cdots X_{y_n}^o T$.
(we use boxes to denote the variables used in each derivation step):
Now,

$$S' \to^* T_1 X_{y_1}^o X_{y_2}^o \cdots \boxed{X_{y_n}^o T} \to \boxed{T_1 X_{y_1}^o} X_{y_2}^o \cdots T_4 X_{y_n}^o \qquad (\text{ by } P_N(6))$$

$$\to^* X_{y_1}^N X_{y_2}^N \cdots X_{y_{n-1}}^N T_1 T_4 X_{y_n}^o \qquad (\text{ by } P_N(3))$$

$$= X_{y_1}^N X_{y_2}^N \cdots X_{y_{n-1}}^N \boxed{Z'' X_{y_n}^o}$$

$$\to X_{y_1}^N X_{y_2}^N \cdots X_{y_{n-1}}^N \boxed{X_{y_n}^D Z} \qquad (\text{ by } P_N(10))$$

$$\to^* \theta(y_1)\theta(y_2)\cdots\theta(y_{n-1})y_n \qquad (\text{ by } P_N(11), P_N(7) \text{ and } P_N(9))$$

So, for $i = n - 1$, $x \in L(G_N)$. Now,

$$S' \to^* \boxed{T_1 X_{y_1}^o} X_{y_2}^o \cdots X_{y_n}^o T_2 \to^* X_{y_1}^N \cdots X_{y_n}^N T_1 T_2 \qquad (\text{ by } P_N(3))$$

$$= X_{y_1}^N \cdots \boxed{X_{y_n}^N Z'}$$

$$\to^* \theta(y_1)\cdots\theta(y_n) \qquad (\text{ by } P_N(8) \text{ and } P_N(9))$$

So, for $i = n$, $x \in L(G_N)$. Now,

$$S' \to^* \boxed{T_1 X_{y_1}^o} X_{y_2}^o \cdots X_{y_n}^o T_2$$

$$\to X_{y_1}^N \boxed{T_1 X_{y_2}^o} \cdots X_{y_n}^o T_2 \qquad \text{(by } P_N(3))$$

$$\to^* X_{y_1}^N \cdots X_{y_i}^N T_1 X_{y_{i+1}}^o \cdots \boxed{X_{y_n}^o T_2} \qquad \text{(by } P_N(3))$$

$$\to X_{y_1}^N \cdots X_{y_i}^N T_1 X_{y_{i+1}}^o \cdots \boxed{X_{y_{n-1}}^o T_3} X_{y_n}^F \qquad \text{(by } P_N(4))$$

$$\to^* X_{y_1}^N \cdots X_{y_i}^N T_1 T_3 X_{y_{i+1}}^o \cdots X_{y_n}^F \qquad \text{(by } P_N(5))$$

$$= X_{y_1}^N \cdots X_{y_i}^N \boxed{Z X_{y_{i+1}}^o} X_{y_{i+2}}^o \cdots X_{y_n}^F$$

$$\to X_{y_1}^N \cdots X_{y_i}^N \boxed{Z X_{y_{i+1}}^I X_{y_{i+2}}^o} \cdots X_{y_n}^F \qquad \text{(by } P_N(12))$$

$$\to X_{y_1}^N \cdots X_{y_i}^N X_{y_{i+1}}^I \boxed{Z X_{y_{i+2}}^R X_{y_{i+3}}^o} \cdots X_{y_n}^F \qquad \text{(by } P_N(13))$$

$$\to X_{y_1}^N \cdots X_{y_i}^N X_{y_{i+1}}^I X_{y_{i+2}}^R \boxed{Z X_{y_{i+3}}^R X_{y_{i+4}}^o} \cdots X_{y_n}^F \qquad \text{(by } P_N(14))$$

$$\to^* X_{y_1}^N \cdots X_{y_i}^N X_{y_{i+1}}^I X_{y_{i+2}}^R X_{y_{i+3}}^R X_{y_{i+4}}^R \cdots \boxed{Z X_{y_{n-1}}^R X_{y_n}^F} \qquad \text{(by } P_N(14))$$

$$\to X_{y_1}^N \cdots X_{y_i}^N X_{y_{i+1}}^I X_{y_{i+2}}^R \cdots \boxed{X_{y_{n-2}}^R X_{y_n}^T Z} X_{y_{n-1}}^F \qquad \text{(by } P_N(15))$$

$$\to X_{y_1}^N \cdots X_{y_i}^N X_{y_{i+1}}^I X_{y_{i+2}}^R \cdots \boxed{X_{y_{n-3}}^R X_{y_n}^T Z} X_{y_{n-2}}^o X_{y_{n-1}}^F \qquad \text{(by } P_N(16))$$

$$\to^* X_{y_1}^N \cdots X_{y_i}^N \boxed{X_{y_{i+1}}^I X_{y_n}^T Z} X_{y_{i+2}}^o \cdots X_{y_{n-2}}^o X_{y_{n-1}}^F \qquad \text{(by } P_N(16))$$

$$\to X_{y_1}^N \cdots X_{y_i}^N X_{y_n}^D \boxed{Z X_{y_{i+1}}^I X_{y_{i+2}}^o} \cdots X_{y_{n-2}}^o X_{y_{n-1}}^F \qquad \text{(by } P_N(17))$$

$$\vdots$$

$$\to^* X_{y_1}^N \cdots X_{y_i}^N X_{y_n}^D X_{y_{n-1}}^D \cdots \boxed{Z X_{y_{i+1}}^I X_{y_{i+2}}^F} \qquad \text{(by prev. derivation)}$$

$$\to X_{y_1}^N \cdots X_{y_i}^N X_{y_n}^D X_{y_{n-1}}^D \cdots X_{y_{i+2}}^D \boxed{X_{y_{i+1}}^D Z} \qquad \text{(by } P_N(18))$$

$$\to X_{y_1}^N \cdots X_{y_i}^N X_{y_n}^D X_{y_{n-1}}^D \cdots \boxed{X_{y_{i+2}}^D Z} y_{i+1} \qquad \text{(by } P_N(11))$$

$$\to^* X_{y_1}^N \cdots \boxed{X_{y_i}^N Z} y_n y_{n-1} \cdots y_{i+2} y_{i+1} \qquad \text{(by } P_N(11))$$

$$\to^* \boxed{Z} \theta(y_1) \cdots \theta(y_i) y_n y_{n-1} \cdots y_{i+2} y_{i+1} \qquad \text{(by } P_N(7))$$

$$\to \theta(y_1) \cdots \theta(y_i) y_n y_{n-1} \cdots y_{i+2} y_{i+1} \qquad \text{(by } P_N(9))$$

Thus, for $i < n - 1$, $x \in L(G_N)$. Therefore, $L' \subseteq L(G_N)$.

Claim 2 : $L(G_N) \subseteq L'$: Consider $y' \in L(G_N)$. We show that y' is in form $\theta(u^R)v^R$ where $uv \in L^R$.
Now, S' derives y'. Then, by $P_N(1)$ and $P_N(2)$, for some $\alpha \in L^R$, either $S' \to^* T_1 h(\alpha) T_2 \to^* y'$ or

$S' \to^* T_1 h(\alpha) T \to^* y'$. Consider $\alpha = a_1 a_2 \cdots a_n$ where each $a_j \in \Sigma$. Then, by definition of h, S' derives

$$T_1 X^o_{a_1} X^o_{a_2} \cdots X^o_{a_n} T_2 \text{ or } T_1 X^o_{a_1} X^o_{a_2} \cdots X^o_{a_n} T. \tag{2}$$

- $S' \to^* T_1 X^o_{a_1} X^o_{a_2} \cdots X^o_{a_n} T$: Then, it generates exactly the following word:

$$\begin{aligned}
S' &\to^* T_1 X^o_{a_1} X^o_{a_2} \cdots X^o_{a_n} T \\
&\to^* X^N_{a_1} X^N_{a_2} \cdots X^N_{a_{n-1}} T_1 T_4 X^o_{a_n} && (\text{by } P_N(3) \& P_N(6)) \\
&= X^N_{a_1} X^N_{a_2} \cdots X^N_{a_{n-1}} \boxed{Z'' X^o_{a_n}} \\
&\to X^N_{a_1} X^N_{a_2} \cdots X^N_{a_{n-1}} X^D_{a_n} Z && (\text{by } P_N(10)) \\
&\to^* Z \theta(a_1) \theta(a_2) \cdots \theta(a_{n-1}) a_n && (\text{by } P_N(11) \& P_N(7)) \\
&\to \theta(a_1) \theta(a_2) \cdots \theta(a_{n-1}) a_n && (\text{by } P_N(9))
\end{aligned}$$

Then, $y' = \theta(a_{n-1} \cdots a_2 a_1) a_n \in L'$ where $a_1 a_2 \cdots a_n \in L^R$.

- $S' \to^* T_1 X^o_{a_1} X^o_{a_2} \cdots X^o_{a_n} T_2$: Then, by applying $P_N(3)$, $P_N(4)$, $P_N(5)$, we have

$$\begin{aligned}
S' \to^* T_1 X^o_{a_1} X^o_{a_2} \cdots X^o_{a_n} T_2 &\to^* X^N_{a_1} X^N_{a_2} \cdots X^N_{a_i} T_1 T_3 X^o_{a_{i+1}} \cdots X^o_{a_{n-1}} X^F_{a_n} \\
&= X^N_{a_1} X^N_{a_2} \cdots X^N_{a_i} Z X^o_{a_{i+1}} \cdots X^o_{a_{n-1}} X^F_{a_n}
\end{aligned}$$

Now, we have the following cases:

- $i = n$: Then,

$$\begin{aligned}
S' \to^* X^N_{a_1} X^N_{a_2} \cdots X^N_{a_n} T_1 T_2 &= X^N_{a_1} X^N_{a_2} \cdots X^N_{a_n} Z' \\
&\to^* Z' \theta(a_1) \theta(a_2) \cdots \theta(a_n) && (\text{by } P_N(8)) \\
&\to \theta(a_1) \theta(a_2) \cdots \theta(a_n) && (\text{by } P_N(9))
\end{aligned}$$

Then, $y' = \theta(a_n \cdots a_2 a_1) \in L'$ where $a_1 a_2 \cdots a_n \in L^R$.

- $i = n - 1$: Then, as $Z = T_1 T_3$,

$$S' \to^* X^N_{a_1} X^N_{a_2} \cdots X^N_{a_{n-1}} Z X^F_{a_n}$$

which can not derive a sequence of terminals.

- $i < n - 1$: Then,

$$S' \to^* X^N_{a_1} X^N_{a_2} \cdots X^N_{a_i} Z X^o_{a_{i+1}} \cdots X^o_{a_{n-1}} X^F_{a_n}.$$

Now, we can apply one of $P_N(7)$, $P_N(9)$ and $P_N(12)$. If we apply $P_N(7)$ or $P_N(9)$, then we can not get a string of terminals. Thus, by applying $P_N(12)$, we have

$$S' \to^* X^N_{a_1} X^N_{a_2} \cdots X^N_{a_i} Z X^I_{a_{i+1}} X^o_{a_{i+2}} \cdots X^o_{a_{n-1}} X^F_{a_n}$$

Again, if we apply $P_N(7)$ or $P_N(9)$, then we can not get a string of terminals. So, we have to apply $P_N(13)$. Let $|X_{a_{i+1}}^I X_{a_{i+2}}^o \cdots X_{a_{n-1}}^o X_{a_n}^F| = m \geq 2$. We now use induction on m to prove that

$$ZX_{a_{i+1}}^I X_{a_{i+2}}^o \cdots X_{a_{n-1}}^o X_{a_n}^F \rightarrow^* Za_n a_{n-1} \cdots a_{i+1}.$$

For $m = 2$,

$$ZX_{a_{n-1}}^I X_{a_n}^F \rightarrow X_{a_n}^D X_{a_{n-1}}^D Z \rightarrow^* Za_n a_{n-1}.$$

For $m = 3$,

$$\boxed{ZX_{a_{n-2}}^I X_{a_{n-1}}^o} X_{a_n}^F \rightarrow X_{a_{n-2}}^I \boxed{ZX_{a_{n-1}}^R X_{a_n}^F} \qquad \text{(by } P_N(13))$$

$$\rightarrow \boxed{X_{a_{n-2}}^I X_{a_n}^T Z} X_{a_{n-1}}^F \qquad \text{(by } P_N(15))$$

$$\rightarrow X_{a_n}^D \boxed{ZX_{a_{n-2}}^I X_{a_{n-1}}^F} \qquad \text{(by } P_N(17))$$

$$\rightarrow X_{a_n}^D X_{a_{n-1}}^D X_{a_{n-2}}^D Z \qquad \text{(by } P_N(18))$$

$$\rightarrow^* Za_n a_{n-1} a_{n-2} \qquad \text{(by } P_N(11))$$

Let the statement be true for strings with lengths less than m. Now,

$$\boxed{ZX_{a_{i+1}}^I X_{a_{i+2}}^o} \cdots X_{a_{n-1}}^o X_{a_n}^F$$

$$\rightarrow X_{a_{i+1}}^I \boxed{ZX_{a_{i+2}}^R X_{a_{i+3}}^o} \cdots X_{a_{n-1}}^o X_{a_n}^F \qquad \text{(by } P_N(13))$$

$$\rightarrow X_{a_{i+1}}^I X_{a_{i+2}}^R ZX_{a_{i+3}}^R \cdots X_{a_{n-1}}^o X_{a_n}^F \qquad \text{(by } P_N(14))$$

$$\rightarrow^* X_{a_{i+1}}^I X_{a_{i+2}}^R X_{a_{i+3}}^R \cdots \boxed{ZX_{a_{n-1}}^R X_{a_n}^F} \qquad$$

$$\rightarrow X_{a_{i+1}}^I X_{a_{i+2}}^R X_{a_{i+3}}^R \cdots \boxed{X_{a_{n-2}}^R X_{a_n}^T Z} X_{a_{n-1}}^F \qquad \text{(by } P_N(15))$$

$$\rightarrow X_{a_{i+1}}^I X_{a_{i+2}}^R X_{a_{i+3}}^R \cdots X_{a_{n-3}}^R X_{a_n}^T ZX_{a_{n-2}}^o X_{a_{n-1}}^F \qquad \text{(by } P_N(16))$$

Here, we can apply one of $P_N(9)$, $P_N(12)$ and $P_N(16)$. If we apply $P_N(9)$ or $P_N(12)$, then S' does not derive a string of terminals. Thus, we apply $P_N(16)$. Then,

$$ZX_{a_{i+1}}^I X_{a_{i+2}}^o \cdots X_{a_{n-1}}^o X_{a_n}^F \rightarrow^* X_{a_{i+1}}^I X_{a_{i+2}}^R X_{a_{i+3}}^R \cdots X_{a_{n-4}}^R X_{a_n}^T ZX_{a_{n-3}}^o X_{a_{n-2}}^o X_{a_{n-1}}^F$$

Now, similar to the above, we can not apply $P_N(9)$ or $P_N(12)$ here. So, we apply $P_N(16)$ and continue the process. Then, we have

$$ZX_{a_{i+1}}^I X_{a_{i+2}}^o \cdots X_{a_{n-1}}^o X_{a_n}^F \rightarrow^* \boxed{X_{a_{i+1}}^I X_{a_n}^T Z} X_{a_{i+2}}^o \cdots X_{a_{n-2}}^o X_{a_{n-1}}^F$$

$$\rightarrow X_{a_n}^D ZX_{a_{i+1}}^I X_{a_{i+2}}^o \cdots X_{a_{n-2}}^o X_{a_{n-1}}^F \qquad \text{(by } P_N(17))$$

$$\rightarrow^* X_{a_n}^D Za_{n-1} \cdots a_{i+1} \qquad \text{(by induction)}$$

$$\rightarrow Za_n a_{n-1} \cdots a_{i+1} \qquad \text{(by } P_N(11))$$

Therefore,

$$
\begin{aligned}
S' \to^* \ & X_{a_1}^N X_{a_2}^N \cdots X_{a_i}^N Z X_{a_{i+1}}^I X_{a_{i+2}}^o \cdots X_{a_{n-1}}^o X_{a_n}^F \\
\to^* \ & X_{a_1}^N X_{a_2}^N \cdots X_{a_i}^N Z a_n a_{n-1} \cdots a_{i+1} \\
\to^* \ & \theta(a_1)\theta(a_2)\cdots\theta(a_i) a_n a_{n-1}\ldots a_{i+1} \qquad\qquad \text{(by } P_N(7) \text{ and } P_N(9)\text{)}.
\end{aligned}
$$

Then, $y' = \theta(a_i a_{i-1}\cdots a_2 a_1) a_n a_{n-1}\ldots a_{i+1} \in L'$ where $a_1 a_2 \cdots a_n \in L^R$.

Therefore, $L(G_N) \subseteq L'$. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ □

A3. A language L is conjugate-free if for all $w \in L$, $C(w) \cap L = \{w\}$. It is undecidable to determine whether or not a given context-free language L is conjugate-free.

Proof: Let Σ be alphabet and (U, V) be an instance of PCP problem where $U = (u_0, u_1, \ldots, u_{n-1})$, $V = (v_0, v_1, \ldots, v_{n-1})$ with each $u_i, v_i \in \Sigma^*$. Assume that the symbols $X, Y, a_0, a_1, \cdots, a_{n-1}$ are distinct and not in Σ. Consider $\Sigma' = \Sigma \cup \{X, Y, a_0, a_1, \cdots, a_{n-1}\}$. We define a context-free grammar $G = (N, \Sigma', P, S)$ where $N = \{S, A, B\}$ is the set of non-terminals, Σ' is the set of terminals, S is the start symbol, and P is the set of derivation rules which contains the following rules:

- $S \to XA \mid YB$

- $A \to u_0 A a_0 \mid u_1 A a_1 \mid \cdots \mid u_{n-1} A a_{n-1} \mid Y$

- $B \to a_0 B v_0 \mid a_1 B v_1 \mid \cdots \mid a_{n-1} B v_{n-1} \mid X$

Now, $L(G)$ denotes the set of all words generated by the grammar G and any string of $L(G)$ is in one of the following forms :

$$
X u_{j_1} u_{j_2} \cdots u_{j_l} Y a_{j_l} \cdots a_{j_2} a_{j_1} \tag{i}
$$

or

$$
Y a_{j_1} a_{j_2} \cdots a_{j_l} X v_{j_l} \cdots v_{j_2} v_{j_1}. \tag{ii}
$$

We now show that $L(G)$ is not conjugate-free if and only if the PCP instance (U, V) has a solution. First, we prove that if the PCP instance (U, V) has a solution, then $L(G)$ is not conjugate-free. Assume that the PCP instance (U, V) has a solution. Then, there is a sequence of integers $i_0, i_1, \ldots, i_k$ such that $u_{i_0} u_{i_1} \cdots u_{i_k} = v_{i_0} v_{i_1} \cdots v_{i_k}$. Now, $L(G)$ contains the following elements:

$$
\begin{aligned}
\beta' &= Y a_{i_k} a_{i_{k-1}} \cdots a_{i_0} X v_{i_0} \cdots v_{i_{k-1}} v_{i_k} = Y a_{i_k} a_{i_{k-1}} \cdots a_{i_0} X u_{i_0} u_{i_1} \cdots u_{i_k} \ \text{ and} \\
\alpha' &= X u_{i_0} u_{i_1} \cdots u_{i_k} Y a_{i_k} \cdots a_{i_1} a_{i_0}.
\end{aligned}
$$

Clearly, $\beta' \in C(\alpha')$. Since $X \neq Y$, $\alpha' \neq \beta'$. Thus, $L(G)$ is not conjugate-free.

Now, we prove the other part, i.e., if $L(G)$ is not conjugate-free, then the PCP instance (U, V) has a solution. Assume that $L(G)$ is not conjugate-free, i.e., there exist w_1, w_2 in $L(G)$ such that $w_2 \in C(w_1)$ and $w_1 \neq w_2$. Then, $w_1 = xy, w_2 = yx$ for some $x, y \in \Sigma'^+$. Since elements of $L(G)$ are in the form (i) or (ii), we have two cases:

- **Case I :** Consider $w_1 = xy$ is in the form (i). Then, for some integer m,

$$w_1 = xy = X u_{i_1} u_{i_2} \cdots u_{i_m} Y a_{i_m} \cdots a_{i_2} a_{i_1}.$$

Now, $\{X, Y, a_0, a_1, \cdots, a_{n-1}\} \cap \Sigma = \emptyset$, each $u_i \in \Sigma^*$, and $x, y \in \Sigma'^+$. Then as $w_2 \in C(w_1)$, w_2 must be in the form (ii). Then, for some integer r,

$$w_2 = yx = Y a_{j_r} a_{j_{r-1}} \cdots a_{j_1} X v_{j_1} \cdots v_{j_{r-1}} v_{j_r}.$$

Since $|w_1|_Y = 1$, $Y \in Pref(w_2)$ and $w_2 \in C(w_1)$, we have $x = X u_{i_1} u_{i_2} \cdots u_{i_m}$ and $y = Y a_{i_m} \cdots a_{i_2} a_{i_1}$. Then, $w_2 = yx = Y a_{j_r} a_{j_{r-1}} \cdots a_{j_1} X v_{j_1} \cdots v_{j_{r-1}} v_{j_r}$ implies

$$Y a_{i_m} \cdots a_{i_2} a_{i_1} X u_{i_1} u_{i_2} \cdots u_{i_m} = Y a_{j_r} a_{j_{r-1}} \cdots a_{j_1} X v_{j_1} \cdots v_{j_{r-1}} v_{j_r} \tag{3}$$

Using our assumed conditions, we have the following from Equation (3): $r = m$, $i_1 = j_1$, $i_2 = j_2$, $\ldots, i_m = j_m$ and $u_{i_1} u_{i_2} \cdots u_{i_m} = v_{i_1} v_{i_2} \cdots v_{i_m}$. Hence, the PCP instance has a solution.

- **Case II :** Consider $w_1 = xy$ is in the form (ii). Then similar to Case I, we show that the PCP instance has a solution.

Therefore, as PCP is undecidable Post (1946), determining whether or not L is conjugate-free is also undecidable when L is context-free.

$\square$