
Fatoração Lyndon Dinâmica após uma Operação de Substituição

LUÍSA ANDRADE MARTINS



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Uberlândia
2024

LUÍSA ANDRADE MARTINS

**Fatoração Lyndon Dinâmica após uma Operação de
Substituição**

Dissertação de mestrado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Felipe Alves da Louza

Coorientador: Marcelo Keese Albertini

Uberlândia

2024

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

M386 2024	Martins, Luísa Andrade, 1996- Fatoração Lyndon dinâmica após uma operação de substituição [recurso eletrônico] / Luísa Andrade Martins. - 2024. Orientador: Felipe Alves da Louza. Coorientador: Marcelo Keese Albertini. Dissertação (Mestrado) - Universidade Federal de Uberlândia, Pós-graduação em Ciência da Computação. Modo de acesso: Internet. Disponível em: http://doi.org/10.14393/ufu.di.2024.158 Inclui bibliografia. Inclui ilustrações. 1. Computação. I. Louza, Felipe Alves da, 1988-, (Orient.). II. Albertini, Marcelo Keese, 1984-, (Coorient.). III. Universidade Federal de Uberlândia. Pós-graduação em Ciência da Computação. IV. Título. CDU: 681.3
--------------	--

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:

Gizele Cristine Nunes do Couto - CRB6/2091
Nelson Marcos Ferreira - CRB6/3074



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Coordenação do Programa de Pós-Graduação em Ciência da
Computação

Av. João Naves de Ávila, 2121, Bloco 1A, Sala 243 - Bairro Santa Mônica, Uberlândia-MG,
CEP 38400-902

Telefone: (34) 3239-4470 - www.ppgco.facom.ufu.br - cpgfacom@ufu.br



ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Ciência da Computação				
Defesa de:	Dissertação de Mestrado, 7/2024, PPGCO				
Data:	26 de fevereiro de 2024	Hora de início:	10:08	Hora de encerramento:	12:15
Matrícula do Discente:	12112CCP019				
Nome do Discente:	Luísa Andrade Martins				
Título do Trabalho:	Fatoração Lyndon Dinâmica após uma Operação de Substituição				
Área de concentração:	Ciência da Computação				
Linha de pesquisa:	Ciência de Dados				
Projeto de Pesquisa de vinculação:	-----				

Reuniu-se por videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Ciência da Computação, assim composta: Professores Doutores: Marcelo Keese Albertini- FACOM/UFU (Coorientador), Paulo Henrique Ribeiro Gabriel- FACOM/UFU, Guilherme Pimentel Telles- IC/UNICAMP e Felipe Alves da Louza - FEELT/UFU, orientador da candidata.

Os examinadores participaram desde as seguintes localidades: Guilherme Pimentel Telles - Campinas/SP . Os outros membros da banca e a aluna participaram da cidade de Uberlândia.

Iniciando os trabalhos o presidente da mesa, Prof. Dr. Felipe Alves da Louza, apresentou a Comissão Examinadora e a candidata, agradeceu a presença do público, e concedeu à Discente a palavra para a exposição do seu trabalho. A duração da apresentação da Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir a senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir a candidata. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando a candidata:

Aprovada

Esta defesa faz parte dos requisitos necessários à obtenção do título de Mestre.

O competente diploma será expedido após cumprimento dos demais requisitos,

Este trabalho é dedicado aos meus pais que sempre acreditaram em mim e me deram a vida.

Agradecimentos

Agradeço à Deus e meus pais, Simara e Emílio, pela oportunidade de me desenvolver intelectualmente. Agradeço também ao meu companheiro Diogo que me deu suporte constante para o desenvolvimento e finalização do mestrado. Agradeço aos meus irmãos, Pedro Emílio e Laura, por me apoiarem durante o processo. Por fim, agradeço aos meus orientadores, Marcelo e Felipe, pela paciência e apoio durante essa jornada.

*“Crê em ti mesmo, age e verás os resultados. Quando te esforças, a vida também se
esforça para te ajudar.”
(Chico Xavier)*

Resumo

Esta dissertação investiga o problema da atualização da fatoração Lyndon de uma palavra w , de tamanho n , quando uma operação de substituição em um símbolo $w[i]$ é realizada, com $0 \leq i < n$. O algoritmo proposto calcula a fatoração Lyndon da nova palavra w' a partir dos fatores Lyndon de w avaliando o efeito da alteração em tempo $O(n_j)$, em que n_j é o tamanho do fator Lyndon em que $w[i]$ está. Em particular, oito diferentes casos são descritos para obter a fatoração de w' a partir dos dos fatores Lyndon de w . O custo computacional em três casos é $O(n_j)$, enquanto que, nos outros cinco casos, o custo é $O(n)$, isto é, o mesmo custo para calcular a fatoração do de w' com o algoritmo de Duval. Conforme a literatura disponível e consultada, este é o primeiro trabalho que aborda o problema da fatoração Lyndon dinâmica a partir de operações de substituição em w . Os resultados apresentados nesta dissertação visam contribuir para uma nova perspectiva para esse problema.

Palavras-chave: Palavras de Lyndon, Fatoração Lyndon, Combinatória de Palavras, Atualização de Textos, Atualização Fatores Lyndon.

Abstract

This dissertation investigates the problem of updating the Lyndon factorization of a word w , of size n , when a substitution operation is performed on a symbol $w[i]$, with $0 \leq i < n$. The proposed algorithm computes the Lyndon factorization of the new word w' from the Lyndon factors of w by assessing the effect of the change in time $O(n_j)$, where n_j is the size of the Lyndon factor in which $w[i]$ is located. In particular, eight different cases are described to obtain the factorization of w' from the Lyndon factors of w . The computational cost in three cases is $O(n_j)$, whereas, the other five cases, the cost is $O(n)$, that is, the same cost to calculate the factorization of w' with Duval's algorithm. According to the available and consulted literature, this is the first work that addresses the problem of dynamic Lyndon factorization from substitution operations in w . The results presented in this dissertation aim to contribute to a new perspective on this problem.

Keywords: Lyndon Words, Lyndon Factorization, Word Combinatorics, Text Updating, Lyndon Factors Updating.

Sumário

1	INTRODUÇÃO	10
2	CONCEITOS FUNDAMENTAIS	13
2.1	Definições	13
2.2	Fatoração Lyndon	16
3	TRABALHOS CORRELATOS	19
3.1	Algoritmo de Duval	19
3.2	LF-SKIP	23
4	ALGORITMO PROPOSTO	27
4.1	Atualização dos fatores Lyndon	27
4.1.1	Alteração na início de um fator	29
4.1.2	Alteração em qualquer posição de um fator	36
4.1.3	Considerações Finais	43
5	CONCLUSÃO	45
	REFERÊNCIAS	47

CAPÍTULO 1

Introdução

As palavras Lyndon (CHEN; FOX; LYNDON, 1958), nomeadas em homenagem ao matemático Roger Lyndon, desempenham um papel importante na área de Combinatória de Palavras (LOTHAIRE, 1997; CROCHEMORE; RYTTER, 2002; SMYTH, 2003). Uma palavra é do tipo Lyndon quando ela é estritamente menor, em ordem lexicográfica, do que todas as suas rotações. Qualquer palavra pode ser decomposta de forma única em subpalavras (ou fatores) Lyndon, tal que esses fatores são dados em ordem não crescente. Essa decomposição, conhecida como fatoração Lyndon canônica, ou padrão, pode ser realizada em tempo linear (DUVAL, 1983).

A fatoração Lyndon de uma palavra pode ser utilizada em muitas aplicações que envolvem o processamento de textos, como em Recuperação de Informação (NEUBURGER; SOKOL, 2013; MANTACI et al., 2014; BANNAI et al., 2019), Compressão de Dados (I et al., 2013; BANNAI et al., 2017; TSURUTA et al., 2020) e Biologia Computacional (DELGRANGE; RIVALS, 2004; BANNAI et al., 2022; BONIZZONI et al., 2022).

No campo da Biologia Computacional, por exemplo, a análise genética tem testemunhado avanços notáveis, sobretudo com a introdução de tecnologias de sequenciamento de última geração. Iniciativas como o projeto 1000 GENOMES¹ e UK10K² têm estabelecido padrões referenciais, fornecendo um rico acervo de dados e *insights*. Esses projetos utilizam genomas de referência, uma representação linear do genoma de uma

¹ <<https://www.internationalgenome.org/>>

² <<https://www.uk10k.org/>>

espécie, simplificando a análise de variações genéticas entre indivíduos. Uma faceta crucial dessa análise é a representação de um genoma de referência, especialmente o desafio de representá-lo como uma sequência indeterminada, onde posições específicas podem assumir múltiplos valores. Tais generalizações são vitais, considerando-se as ambiguidades e erros frequentemente encontrados em sequências de DNA e proteínas.

A fatoração Lyndon de uma palavra w depende da ordenação lexicográfica de seus sufixos (GLEN; SIMPSON; SMYTH, 2017). Dessa forma, operações de edição em símbolos de w podem modificar significativamente o resultado da fatoração Lyndon da palavra modificada w' , o que leva à seguinte pergunta:

É possível obter a fatoração Lyndon depois de editar w sem a necessidade de recalcular todos os fatores de w' ?

Nesta dissertação, foi considerada uma variação do problema acima em que apenas operações de substituição são permitidas na edição de w . Neste cenário, o tamanho de w não é alterado, no entanto, a sua fatoração pode ser completamente diferente após a edição. Esse tipo de fatoração dinâmica pode ser aplicada, por exemplo, no desenvolvimento de um índice dinâmico baseado na Transformada de Burrows-Wheeler Bijetiva (BANNAI et al., 2019).

Conforme a literatura disponível e consultada, este é o primeiro trabalho na literatura que aborda essa questão para fatoração Lyndon dinâmica.

O algoritmo apresentado no Capítulo 4 avalia o efeito da alteração no símbolo $w[i]$ na fatoração Lyndon em tempo $O(n_j)$, em que n_j é o tamanho do fator Lyndon em que $w[i]$ está. Em particular, oito diferentes casos são descritos para atualizar os fatores de w e obter a fatoração Lyndon de w' . O custo computacional de tempo em três casos é $O(n_j)$, enquanto que, para os outros cinco casos, o custo é $O(n)$, isto é, o mesmo custo para calcular a fatoração de w' com o algoritmo de (DUVAL, 1983). Como questão em aberto fica se é possível obter a fatoração Lyndon de w' em tempo $o(n)$ utilizando estruturas de dados auxiliares, como o vetor de sufixos e o vetor de maior prefixo em comum (MANBER; MYERS, 1993). Os resultados apresentados neste trabalho visam contribuir para uma nova perspectiva para este problema.

Esta dissertação está organizada da seguinte forma: O Capítulo 2 apresenta as definições e teorias fundamentais sobre o tema. O Capítulo 3 explora estudos anteriores e metodologias relevantes. O Capítulo 4 apresenta os resultados desenvolvidos para

a atualização dos fatores Lyndon. Por fim, o Capítulo 5 conclui o texto e apresenta perspectivas para trabalhos futuros.

Conceitos Fundamentais

2.1 Definições

Seja um alfabeto Σ um conjunto finito com σ elementos chamados de símbolos, letras, ou caracteres.

Definição 2.1.0.1. *Uma palavra s , de tamanho $|s| = n$, é uma sequência finita de símbolos de Σ justapostos, tal que o i -ésimo símbolo de s é denotado por $s[i]$, com $0 \leq i < n$.*

O conjunto de todas as palavras não vazias que podem ser geradas com símbolos de Σ é denotado por Σ^+ . A palavra vazia é denotada por ε e $\Sigma^* = \Sigma^+ \cup \varepsilon$.

Definição 2.1.0.2. *Uma subpalavra de s que começa na posição i e termina na posição j é denotada por $s[i, j]$, com $0 \leq i, j < n$.*

Por exemplo, a palavra $s = aabaaabaa$ possui as seguintes subpalavras.

$$\left[\begin{array}{c} a, a, b, a, a, a, b, a, \\ aa, ab, ba, aa, aa, ab, ba, \\ aab, aba, baa, aaa, aab, aba, \\ aaba, abaa, baaa, aaab, aaba, \\ aabaa, abaaa, baaab, aaaba, \\ aabaaa, abaaab, baaaba, \\ aabaaab, abaaaba, \\ aabaaaba \end{array} \right]$$

Definição 2.1.0.3. Um prefixo de s é uma subpalavra do tipo $s[0, i]$, enquanto que um sufixo é uma subpalavra do tipo $s[j, n - 1]$, com $0 \leq i, j < n$. Em particular, $s[0, i]$ (respectivamente $s[j, n - 1]$) é chamada de prefixo (respectivamente sufixo) próprio de s se $i \neq n - 1$ (respectivamente $j \neq 0$). Um sufixo $s[j, n - 1]$ será denotado por $s[j, ..]$.

Por exemplo, a palavra $s = aabaaabaa$, possui os seguintes sufixos próprios.

$$\begin{bmatrix} abaaabaa, \\ baaabaa, \\ aaabaa, \\ aabaa, \\ abaa, \\ baa, \\ aa, \\ a \end{bmatrix}$$

Definição 2.1.0.4. Uma palavra w é um fator (ou subpalavra) de s se $s = uwv$, com $u, v \in \Sigma^*$. Em particular, $|w| = m$ é um fator de $|s| = n$ se $w[0, m - 1] = s[i, i + m - 1]$ para algum $i \leq n - m$.

Por exemplo, $w = aa$ é um fator da palavra $s = aabaaabaa$, para os valores de i iguais à 0, 3, 4 e 7.

Definição 2.1.0.5. A concatenação de duas palavras s e w corresponde à palavra u composta pelos símbolos de s seguidos pelos símbolos de w , denotado por $u = s \cdot w$. A concatenação k vezes da mesma palavra é denotada por $u^k = \underbrace{u \cdot u \cdots u}_k$.

Definição 2.1.0.6. Uma palavra $w \in \Sigma^+$ é chamada de primitiva se não puder ser escrita como s^k , com $s \in \Sigma^+$ e $k \geq 2$, isto é, w não é uma concatenação repetida de uma outra palavra.

Por exemplo, a palavra $u = aabaaabaa$ é primitiva, enquanto que $v = aabaaaba$ não é primitiva, uma vez que $v = w^2$, para $w = aaba$.

Definição 2.1.0.7. Uma palavra $w = w[0, n - 1]$ é um conjugado (ou rotação cíclica) de $s = s[0, n]$ se $w[0, n - 1] = s[i, n - 1]s[0, i - 1]$ para algum $0 < i \leq n$.

Por exemplo, a palavra $s = aabaaabaa$, possui os seguintes conjugados:

$$\begin{bmatrix} aabaaabaa, \\ abaaabaaa, \\ baaabaaaa, \\ aaabaaaab, \\ aabaaaaba, \\ abaaaabaa, \\ baaaabaaa, \\ aaaabaaab, \\ aaabaaaba \end{bmatrix}$$

Definição 2.1.0.8. O período de uma palavra w é um inteiro p , com $0 \leq p \leq |w|$, tal que $w[i] = w[i + p]$ para todo $i \in \{0, \dots, |s| - p - 1\}$. Quando não há ambiguidade, a subpalavra $w[0, p - 1]$ também é chamada de período de w .

Por exemplo, os períodos da palavra $w = aabaaabaa$ são 4, 7, 8 e 9, correspondentes às seguintes subpalavras.

$$\begin{bmatrix} aaba, \\ aabaaab, \\ aabaaaba, \\ aabaaabaa \end{bmatrix}$$

O tamanho do menor período de uma palavra w é denotado por $per(w)$. Por exemplo, para a palavra $w = aabaaabaa$, o valor de $per(w) = 4$.

Note que o tamanho da palavra w é um período, portanto, toda palavra não-vazia tem pelo menos um período.

Definição 2.1.0.9. Uma palavra $w \in \Sigma^+$ é menor (lexicograficamente) do que outra palavra $s \in \Sigma^+$, denotado por $w < s$, se w for prefixo próprio de s ou $w[0, j - 1] = s[0, j - 1]$ e $w[j] < s[j]$, para algum j .

Definição 2.1.0.10. Uma palavra $w \in \Sigma^+$ é maior (lexicograficamente) do que outra palavra $s \in \Sigma^+$, denotado por $w > s$, se s for prefixo próprio de w ou $w[0, j-1] = s[0, j-1]$ e $s[j] < w[j]$, para algum j .

Proposição 2.1.1. Uma palavra $w \in \Sigma^+$ é maior (lexicograficamente) do que todos os seus prefixos próprios.

O tamanho do maior prefixo comum entre duas palavras w e s será denotado por $\text{lcp}(w, s)$. O lcp permite identificar rapidamente a maior subsequência inicial comum entre duas ou mais palavras, facilitando a comparação, a ordenação e a busca de informações em grandes volumes de dados.

Definição 2.1.1.1. Uma palavra $w \in \Sigma^+$ é menor (lexicograficamente) do que outra palavra $s \in \Sigma^+$, denotado por $w < s$ se $\text{lcp}(w, s) = |w| < |s|$ ou se $w[\text{lcp}(w, s) + 1] < s[\text{lcp}(w, s)]$. Da mesma forma, duas palavras são iguais quando $\text{lcp}(w, s) = |w| = |s|$.

Por exemplo, para as palavras $w = aabaaabaa$ e $s = aabbaa$, $\text{lcp}(w, s) = 3$. Além disso, $w < s$, pois $\text{lcp}(w, s) = 3$ e $w[3] < s[3]$.

2.2 Fatoração Lyndon

Definição 2.2.0.1. (CHEN; FOX; LYNDON, 1958) Uma palavra $w \in \Sigma^+$ é Lyndon (ou simples) quando w é menor, em ordem lexicográfica, do que todos os seus conjugados próprios.

Por exemplo, a palavra $w = abababc$ é Lyndon, pois w é menor do que todos os seus conjugados.

$$\begin{bmatrix} bababca, \\ ababcab, \\ babcaba, \\ abcabab, \\ bcababa, \\ cababab \end{bmatrix}$$

No decorrer do texto, será utilizado a notação de justaposição para representar a concatenação e em casos específicos será utilizado $\cdot$ para evidenciar a concatenação.

Proposição 2.2.1. *Uma palavra $w \in \Sigma^+$ é Lyndon se e somente se, $|w| = 1$ ou $w = uv$ tal que $u, v \in \Sigma^+$ são palavras Lyndon e $u < v$.*

Por exemplo, a palavra $w = abac$ é Lyndon, então $ab < ac$, com ab, ac palavras Lyndon.

Alternativamente, uma palavra w é Lyndon se e somente se ela for lexicograficamente menor do que qualquer um de seus sufixos próprios (GLEN; SIMPSON; SMYTH, 2017) . Como exemplo, tome a palavra $aabababc$, que é Lyndon, pois é menor que seus sufixos próprios.

$$\left[\begin{array}{c} abababc, \\ bababc, \\ ababc, \\ babc, \\ abc, \\ bc, \\ c \end{array} \right]$$

Podem existir diferentes formas de dividir (fatorar) uma palavra w em fatores Lyndon. Por exemplo, a palavra $w = abac$ pode ser fatorada como $ab \cdot ac$ e $abac$.

Chen, Fox e Lyndon (CHEN; FOX; LYNDON, 1958) propuseram uma fatoração particular, denotada por $CFL(w)$, chamada de fatoração Lyndon canônica (ou padrão), em que w é fatorada em uma sequência não crescente de palavras Lyndon.

Teorema 2.2.2. *Qualquer palavra w admite uma fatoração Lyndon única*

$$CFL(w) = w_0 \cdot w_1 \cdots w_{k-1}$$

tal que w_i é Lyndon, para todo $i \in \{0, \dots, k-1\}$, onde $w_0 \geq w_1 \geq \dots \geq w_{k-1}$.

Por exemplo, seja $w = cctactg$, a fatoração Lyndon é dada por $CFL(w) = cct \cdot actg$, com $cct \geq actg$.

Note que se w for Lyndon, pela definição $CFL(w) = w$.

Propriedade 2.2.3. *O último fator w_{k-1} de $CFL(w) = w_0 \cdot w_1 \cdots w_{k-1}$ corresponde ao menor sufixo (lexicograficamente) de w .*

Trabalhos Correlatos

Este capítulo explora as contribuições anteriores no domínio da análise e processamento de palavras, focando particularmente nos algoritmos de Duval e LF-Skip. A motivação para se aprofundar nesses dois algoritmos decorre da importância fundamental do algoritmo de Duval e das eficiências computacionais oferecidas pelo LF-Skip, especialmente em contextos com alfabetos reduzidos. Essa característica se mostra especialmente vantajosa em aplicações práticas, como na modificação de um nucleotídeo em uma sequência genética.

3.1 Algoritmo de Duval

Dada uma palavra $w \in \Sigma^+$, de tamanho n , Esse é um algoritmo guloso, isto é, a cada estágio uma decisão ótima localmente é tomada.

Primeiro, será introduzida uma nova notação que será utilizada pelos algoritmos deste capítulo.

Definição 3.1.0.1. *Uma palavra w é chamada de pré-simples, se tem a forma $w = uu\bar{u}$, onde u é uma palavra simples (Lyndon) e $\bar{u}$ é um prefixo de u , que pode ser vazio.*

Note que toda palavra Lyndon também é pré-simples. Por exemplo, $w = abcabcab$ é uma palavra pré-simples, formada pela repetição da palavra simples $u = abc$ duas vezes e depois seguida por um prefixo de u dado por $\bar{u} = ab$.

Esta palavra começa com uma repetição de uma palavra Lyndon e termina com um prefixo da mesma palavra Lyndon, atendendo à definição de uma palavra pré-simples conforme o método proposto por (DUVAL, 1983) e descrito no algoritmo de Duval 1 apresentado adiante.

A cada passo do algoritmo, a palavra w será dividida em três partes: $w = w_0w_1w_2$. A primeira, w_0 , é a parte processada, logo a fatoração Lyndon dessa parte foi calculada anteriormente. A parte w_1 é uma palavra pré-simples contendo $\bar{u}$ e w_2 é o sufixo de w que ainda precisa ser processado. A cada iteração, o primeiro símbolo da palavra w_2 é acessado e testa-se o que ocorre ao adicioná-lo à palavra w_1 . Caso ao adicionar esse símbolo torne a palavra w_1 não pré-simples, w_1 será dividida. A parte que será separada é a subsequência de w_1 que chega até o ponto onde o símbolo adicionado interrompe a pré-simplicidade. Esta subsequência é reconhecida como um fator de Lyndon e é então removida de w_1 . Após ser retirada, ela é acrescentada à sequência de fatores de Lyndon que já foram estabelecidos em w_0 .

Durante a execução do algoritmo, ele tenta expandir w_1 anexando símbolos de w_2 (linhas 6 a 12). Assim, se o símbolo atual em w_2 ($w[j]$) for igual ao símbolo em w_1 que está sendo comparado ($w[k]$), o algoritmo incrementa ambos os índices j e k . Neste caso, w_1 mantém sua pré-simplicidade. Se $w[j]$ for maior que $w[k]$, então ao adicionar $w[j]$ a w_1 , essa parte torna-se uma palavra simples. O índice j é incrementado para incluir esse símbolo em w_1 , e k é reiniciado para comparar o próximo símbolo de w_2 com o início de w_1 . Se $w[j]$ for menor que $w[k]$, isso indica que a palavra $w_1 \cdot w[j]$ não é mais pré-simples. Neste ponto, a parte atual de w_1 (que era pré-simples até agora) é dividida em suas palavras simples constituintes e o restante (que pode ser vazio). A palavra simples identificada terá o comprimento $j - k$. Essa parte é então adicionada à fatoração de Lyndon em w_0 e o algoritmo começa a processar novamente a parte restante de w_1 na próxima iteração (linhas 14 a 16).

Para fins didáticos, o símbolo \$ será utilizado como a representação do fim da palavra.

Por exemplo, considere a palavra $w = banana$. Tem-se que:

1. **Primeira iteração:** $w_0 = \varepsilon$, $w_1 = b$, e $w_2 = anana$, uma vez que $i = 0$ marca o início da palavra w_1 , $j = i + 1 = 1$ o início da palavra w_2 e $k = i = 0$ o caractere que está sendo comparado atualmente. O intuito é adicionar $w[j]$ a w_1 e, para isso,

Algoritmo 1 Algoritmo de Duval

```

1: Função DUVAL( $w$ )
   Entrada: palavra  $w$ 
   Saída: lista de fatores Lyndon de  $w$ 
2:    $i \leftarrow 0, n \leftarrow |w|, \text{fatoração} \leftarrow []$  ▷  $i$  - início do caractere de  $w_1$ 
3:   Enquanto  $i < n$  Faça
4:      $j \leftarrow i + 1, k \leftarrow i$  ▷  $j$  - início do caractere de  $w_2$ 
5:     ▷  $k$  - caractere que está sendo processado
6:     Enquanto  $j < n$  e  $w[k] \leq w[j]$  Faça
7:       Se  $w[k] < w[j]$ 
8:          $k \leftarrow i$ 
9:       Senão
10:         $k \leftarrow k + 1$ 
11:      Fim Se
12:       $j \leftarrow j + 1$ 
13:    Fim Enquanto
14:    Enquanto  $i \leq k$  Faça
15:       $\text{fatoração.inserir}(w(i, i + j - k))$ 
16:       $i \leftarrow j - k$ 
17:    Fim Enquanto
18:  Fim Enquanto
19:  Retorna fatoração
20: Fim Função

```

compará-lo a $w[k]$.

Nesse caso, ao fazer a comparação, $b > a$, isso significa que $w_1 w[j]$ não é mais uma palavra pré-simples, então a palavra pré-simples de w_1 é separada da parte simples, nesse caso o próprio w_1 . Feito isso, a palavra simples é adicionada ao vetor de fatoração, linha 15. Por fim, j é incrementado.

w_0	w_1	w_2						w_0	w_1	w_2				
	$i\ k$	j							k	$i\ j$				
	$\downarrow\downarrow$	$\downarrow$					$\rightarrow$		$\downarrow$	$\downarrow\downarrow$				
	0	1	2	3	4	5			0	1	2	3	4	5
	b	a	n	a	n	a			b	a	n	a	n	a

2. **Segunda iteração:** $w_0 = b, w_1 = a$, e $w_2 = nana$. Ao comparar $w[k]$ com $w[j]$, verifica-se que a palavra $w[k]w[j] = an$ se transforma em simples, pois $a < n$. As-

sim, o índice é incrementado para $j = 3$ mantendo o valor de $k = 1$. Consequentemente, $w_1 = an$ e $w_2 = na\$$. O mesmo processo é repetido e como $w[1] = w[3]$, k e j são incrementados para $k = 2$ e $j = 4$, mantendo a pré-simplicidade de $w_1 = ana$. O processo repete-se até $j = |w|$, ou seja, chega ao fim da palavra. Então, a subpalavra simples $w[i, j - k] = an$ é adicionada ao vetor.

w_0	w_1	w_2					w_0	w_1	w_2				
	i	k	j					i	k	j			
	$\downarrow\downarrow$	$\downarrow$					$\rightarrow$	$\downarrow\downarrow$	$\downarrow$				
0	1	2	3	4	5		0	1	2	3	4	5	
b	a	n	a	n	a		b	a	n	a	n	a	
$\downarrow$													
w_0	w_1		w_2				w_0	w_1		w_2			
	i	k	j					i	k	j			
	$\downarrow$	$\downarrow$	$\downarrow$			$\dots$		$\downarrow$	$\downarrow$	$\downarrow$			
0	1	2	3	4	5		0	1	2	3	4	5	6
b	a	n	a	n	a		b	a	n	a	n	a	\$

3. **Terceira iteração:** O processo é praticamente o mesmo da segunda iteração, exceto pelo fato de que $w_0 = ban$, $w_1 = a$, $w_2 = na$.

w_0			w_1	w_2				w_0					w_1	w_2	
			i	k	j				k					i	j
			$\downarrow\downarrow$	$\downarrow$	$\dots$				$\downarrow$					$\downarrow$	$\downarrow$
0	1	2	3	4	5				0	1	2	3	4	5	6
b	a	n	a	n	a				b	a	n	a	n	a	\$

4. **Quarta iteração:** $w_0 = banan$, $w_1 = a$, $w_2 = \varepsilon$. Desse modo, como $j = |w|$ e w_1 são simples, adiciona-se w_1 ao vetor de fatoração. Ao final, tem-se $i = 6$, portanto $i = |w|$ e é retornado o vetor de fatoração completo.

w_0					w_1	w_2							
					$i\ k$	j							
					$\downarrow\downarrow$	$\downarrow$							
0	1	2	3	4	5	6							
b	a	n	a	n	a	$\$$							

 $\rightarrow$

w_0					w_1	w_2							
0	1	2	3	4	5	6							
b	a	n	a	n	a	$\$$							

3.2 LF-SKIP

Na literatura é possível encontrar algoritmos eficientes para obter a fatoração Lyndon como apresentado em (FRANEK et al., 2016) e (GHUMAN; GIAQUINTA; TARHIO, 2014). Nesta seção é apresentado o algoritmo LF-SKIP, proposto por (GHUMAN; GIAQUINTA; TARHIO, 2019), que também executa em tempo $O(n)$ para uma palavra de tamanho n , porém usa menos operações em casos específicos.

O LF-SKIP pode auxiliar na preparação de dados para ordenação lexicográfica e facilitar buscas eficientes em grandes conjuntos de dados textuais, otimizando tanto a velocidade quanto a eficácia de algoritmos de busca.

A sua vantagem na prática está no uso para palavras de alfabetos pequenos e, para isso, são utilizados oito lemas como fundamentação do algoritmo.

Sejam $\mathcal{L}$ o conjunto de todas as palavras Lyndon e $\mathcal{P}$ o conjunto não-vazio dos prefixos de palavras em $\mathcal{L}$. Tome $P' = \mathcal{P} \cup \{c^k \mid k \geq 2\}$, onde c é o maior símbolo em Σ .

O lema a seguir afirma que a fatoração Lyndon de qualquer palavra w pode ser dada pela concatenação de w_0 com a fatoração Lyndon do sufixo restante w' . De modo que para calcular w_0 , basta encontrar o maior prefixo de w que é uma palavra Lyndon. Assim, esse mesmo processo pode ser feito de forma recursiva para o sufixo restante w' .

Lema 3.2.1. *Seja $w \in \Sigma^+$ e w_0 o maior prefixo de $w = w_0w'$ que pertença a $\mathcal{L}$. Assim, $CFL(w) = w_0 \cdot CFL(w')$.*

Por exemplo, considere um alfabeto $\Sigma = \{a, b, c\}$ e a palavra $w = acbabcab$. O objetivo é encontrar a fatoração Lyndon dessa palavra com base no Lema 3.2.1. Assim, o primeiro passo é encontrar o prefixo mais longo w_0 de w que seja uma palavra Lyndon,

nesse caso, $w_0 = acb$. Feito isso, o passo seguinte é calcular a fatoração Lyndon do sufixo restante $w' = abcab$. É nesse momento em que o processo é repetido, então para $w' = abcab$ tem-se $w_0 = abc$, como o maior prefixo. Consequentemente, o sufixo restante será $w' = ab$. Esse processo é feito iterativamente até obter todos os fatores.

Seja w uma palavra sobre um alfabeto Σ com $CFL(w) = w_0 \cdot w_1 \cdots w_m$ e β o menor símbolo em Σ . Suponha que exista $k \geq 2$, $i \geq 0$ tal que β^k é um prefixo do fator w_i . Se o último símbolo de w não é β , então pelo Teorema 2.2.2 e pelas propriedades das palavras de Lyndon, β^k é um prefixo de cada um dos fatores $w_{i+1}, w_{i+2}, \dots, w_m$. (GHUMAN; GIAQUINTA; TARHIO, 2014). Observe que caso o último símbolo seja β , basta quebrar a palavra de modo que o último fator não seja β .

Essa propriedade, em particular, é explorada pelo algoritmo LF-SKIP que será apresentado mais adiante.

Agora, considere P um conjunto de padrões que se deseja buscar e $Occ_P(w)$ o conjunto das posições iniciais das ocorrências de P em w . Por exemplo, considere $w = aabcacaa$ e $P = \{aa, bc\}$. Logo, $Occ_P(w) = \{0, 2, 6\}$.

Os lemas a seguir fornecem uma maneira estruturada de decompor uma palavra com base em certas propriedades de símbolos e compreender sua fatoração Lyndon.

Lema 3.2.2. *Seja w uma palavra e seja $s = \max\{i \mid w[i] > \beta\} \cup \{-1\}$. Então,*

$$CFL(w) = CFL(w[0, s]) \cdot CFL(\beta^{|w| - 1 - s})$$

Lema 3.2.3. *Seja w uma palavra tal que $\beta\beta$ ocorra nela, e seja $s = \min\{Occ_{\beta\beta}(w)\}$. Então, $CFL(w) = CFL(w[0, s - 1]) \cdot CFL(w[s, |w| - 1])$.*

Por exemplo, considere as palavras $w = abbabcaa$ e $x = accaabbaab$. Para w , a posição máxima i é 5, pois $w[5] = c$, que é maior que $\beta = a$ e, consequentemente, $s = 5$. Usando o Lema 3.2.2:

$$CFL(w) = CFL(w[0, s])CFL(\beta^{|w| - 1 - s}) = CFL(abbab) \cdot CFL(aa).$$

Em contrapartida, para a palavra x , $s = 3$, pois $\beta\beta$ ocorre na posição 3. Então, pelo Lema 3.2.3,

$$CFL(w) = CFL(w[0, s - 1]) \cdot CFL(w[s, |w| - 1]) = CFL(acc)CFL(aabbaab).$$

Lema 3.2.4. *Considere β o menor elemento do seu alfabeto. Seja w uma palavra tal que*

$$a) w[0] = w[1] = \beta$$

$$b) w[|w| - 1] \neq \beta$$

Tome r como menor posição em w tal que $w[r] \neq \beta$ e $P = \{\beta^r c \mid \forall c \in \Sigma, c \leq w[r]\}$. Então,

$$b_0 = \min\{s \in \text{Occ}_P(w) \mid w[s \dots |w| - 1] < w\} \cup \{|w|\} - 1$$

onde b_0 é a última posição do fator w_0 .

Suponha $w = aacbabaabbaaab$, sobre o alfabeto $\Sigma = \{a, b, c\}$. Observe que as condições a) e b) são satisfeitas, portanto, pelo Lema 3.2.4, $r = 2$ e $P = \{aab, aac\}$. Como $\text{Occ}_P(w) = \{0, 6, 11\}$ $s \in \{6, 11\}$ pois todos os elementos do conjunto devem satisfazer a relação $w[s \dots |w| - 1] < w$. Logo, se $b_0 = \min\{6, 11, 12\} - 1 = 5$, então o primeiro fator será $w_1 = aacbab$.

A partir disso, o algoritmo LF-SKIP foi proposto, cuja ideia consiste em saltar símbolos durante a fatoração Lyndon. Para isso é necessário que w satisfaça as condições a) e b) do Lema 3.2.4. Assim, utilizando os demais lemas e supondo que $\beta\beta$ ocorre em w , é possível fatorar $w = u \cdot v$, onde $|u| = \min\{\text{Occ}_{\beta\beta}(w)\}$. Caso os últimos elementos de v sejam iguais à β , basta adicioná-los ao final da fatoração cada elemento como um fator Lyndon.

Considere a palavra $w = abaabaabbaabaa$ como entrada para o Algoritmo ??. As linhas de 1 a 7 têm como objetivo garantir as condições do Lema 3.2.4, então ao final da iteração $w = w[0, e] = abaabaabbaab$, $l = 2$ e $e = 11$, onde l é a variável responsável por armazenar o comprimento do sufixo contendo apenas caracteres $\beta = a$. Na Linha 8, o valor inicial s que corresponde à menor posição tal que $\beta\beta$ ocorra no prefixo $w[0, e]$ usando $\min\{\{\text{Occ}_{\beta\beta}(w)\} \cup \{|w|\}\}$ é encontrado, neste caso, $s = 3$. Dada as condições satisfeitas, aplica-se o algoritmo de Duval para o prefixo $w[0, s - 1]$.

Após estabelecer as definições iniciais, o algoritmo LF-SKIP avança para o cálculo de b_0 , entre as linhas 10 e 27. Primeiro, ele identifica o menor índice r onde o símbolo em w não é β . Esse índice r é usado para delimitar o início da busca por fatores Lyndon no sufixo restante de w , ou seja, $w[s, |w| - 1]$. Em seguida, o algoritmo constrói o conjunto de padrões P , incluindo todas as sequências de β^r seguidas por um símbolo c que seja menor ou igual a $w[r]$. Para o exemplo dado, com $r = 2$, tem-se $P = aaa, aab$.

O algoritmo então itera sobre as ocorrências dos padrões de P em w , procurando posições de ocorrência que satisfaçam duas condições específicas: a subpalavra de w

começando na posição de ocorrência e indo até o fim de w deve ser lexicograficamente menor que w . Assim, deve-se encontrar a maior subsequência contígua que satisfaça essa condição. Com base nessas posições de ocorrência, o algoritmo determina o intervalo s e a contagem de fatores q . Quando essas condições são atendidas, o algoritmo registra o intervalo (s, q) , atualiza as variáveis s e k e extrai o fator Lyndon correspondente.

Por exemplo, o padrão de $P = \{aab\}$ é encontrado na posição 2 de w , o algoritmo verifica se $w[2..13]$ é menor que w . Se sim, essa posição é considerada para determinar o intervalo s e a contagem de fatores q , uma vez que pode haver repetições do padrão encontrado. Esse processo continua até que o algoritmo tenha processado todo o sufixo $w[s, |w| - 1]$ e identificado todos os fatores Lyndon.

Como no início do processo a palavra w foi quebrada removendo os símbolos β e armazenados pela variável l , esses símbolos são adicionados à fatoração, onde cada fator é o próprio símbolo β resultando em um total de l fatores (linhas 34 a 36).

Algoritmo Proposto

4.1 Atualização dos fatores Lyndon

Seja $w = w[1]w[2] \dots w[n]$ uma palavra sobre um alfabeto Σ de tamanho σ .

Considere a seguinte fatora  o Lyndon (CHEN; FOX; LYNDON, 1958) para w :

$$CFL(w) = w_0 \cdot w_1 \cdots w_j \cdots w_{k-1},$$

onde cada w_i    um fator Lyndon de tamanho n_i e $w_0 \geq w_1 \geq \dots \geq w_{k-1}$.

Neste cap  tulo, ser   apresentado como atualizar $CFL(w)$ quando um   nico s  mbolo $w[i]$ de um fator w_j    alterado.

Exemplo 4.1. Considere a palavra $w = deedecdcdbddbdbdadadaaadaea$, cuja fatora  o Lyndon    dada por:

$$CFL(w) = dee \cdot \underbrace{de}_{w_1} \cdot cd \cdot cd \cdot \underbrace{bdd}_{w_4} \cdot bd \cdot bd \cdot ad \cdot ad \cdot aaadae \cdot a$$

Agora, suponha que o valor de $w[9] = \mathbf{b}$ do fator w_4 seja alterado para $w'[9] = \mathbf{c}$. Assim, a nova palavra $w' = deedecdcddbdbdadadaaadaea$ ter   a seguinte fatora  o Lyndon:

$$CFL(w') = dee \cdot de \cdot \underbrace{cdcd\mathbf{c}dd}_{\text{novo fator}} \cdot bd \cdot bd \cdot ad \cdot ad \cdot aaadae \cdot a$$

Observe que os fatores que antecedem $w_1 = de$ e que sucedem $w_5 = bd$ em $CFL(w)$ s  o mantidos em $CFL(w')$. Isto acontece por que w_1    **maior** do que novo fator criado

$w_2w_3w'_4 = cdcdd$ e uma vez que os fatores w_i que antecedem w_1 , $0 \leq i < 1$, são maiores ou iguais a w_1 , esses também serão maiores do que $cdcd$. O mesmo argumento vale, de forma simétrica, para os fatores que sucedem w_5 .

De forma geral, considerando a atualização de um símbolo $w[i]$ em um fator w_j de $CFL(w)$, uma vez encontrados os fatores w_a e w_b em $CFL(w)$ que delimitam à esquerda e à direita, respectivamente, os fatores Lyndon que devem ser atualizados para obter $CFL(w')$, com $0 \leq a < j < b < k$, é possível definir a nova fatoração para w' da seguinte forma:

$$CFL(w') = w_0 \cdots w_a \cdot CFL(w_{a+1}w_{a+2} \cdots w'_j \cdots w_{b-1}) \cdot w_b \cdots w_{k-1} \quad (1)$$

Serão consideradas as seguintes propriedades da fatoração Lyndon que devem ser mantidas em $CFL(w')$ sempre que um símbolo de um fator w_j for atualizado, gerando um novo fator w'_j .

- (I) **Intrafator:** o fator w'_j deve ser menor (lexicograficamente) do que todos os seus sufixos próprios, isto é,

$$w'_j[0, \dots] < w'_j[\ell, \dots], \text{ para todo } \ell = 1, 2, \dots, n_j - 1$$

- (II) **Interfatores:** os fatores $w_0, w_1, \dots, w_{j-1}$ devem ser maiores ou iguais a w'_j , e os fatores $w_{j+1}, w_{j+2}, \dots, w_{k-1}$ devem ser menores ou iguais a w'_j , isto é,

$$w_0 \geq w_1 \geq \cdots \geq w_{j-1} \geq w'_j \geq w_{j+1} \geq \cdots \geq w_{k-1}$$

Portanto, sempre que ocorrer uma substituição em um símbolo $w[i]$ será preciso verificar se as propriedades (I) e (II) continuam válidas. Dessa forma, quatro casos podem acontecer:

1. As propriedades (I) e (II) são satisfeitas.
2. A propriedade (I) é satisfeita e (II) não é satisfeita.
3. A propriedade (I) não é satisfeita e (II) é satisfeita.
4. As propriedades (I) e (II) não são satisfeitas.

Em particular, serão considerados dois casos de acordo com a posição em que $w[i]$ está em seu fator w_j :

- O símbolo atualizado $w[i]$ está na primeira posição de um fator w_j .
- O símbolo atualizado $w[i]$ está em uma posição p de um fator w_j , com $0 < p < n_j$.

Dessa forma, quando for mencionado que a modificação aconteceu no início do fator, isso se refere especificamente à alteração na primeira posição desse fator.

4.1.1 Alteração na início de um fator

Considere que a posição do símbolo $w[i]$ alterado corresponde ao início de um fator Lyndon w_j de $CFL(w)$, portanto a $w'_j = w'[i]w_j[1, \dots]$.

Inicialmente, será mostrado como verificar as propriedades (I) e (II) após a alteração de $w[i]$ em tempo $O(n_j)$, com $|w_j| = n_j$.

Lema 4.1.1. *Considerando que o símbolo modificado está no início do fator w_j de $CFL(w)$, é possível verificar a propriedade (I) em tempo $O(n_j)$, com $|w_j| = n_j$.*

Demonstração. Caso $w'[i] < w[i]$, a propriedade (I) é trivialmente satisfeita. Esse caso pode ser verificado em tempo $O(1)$. Caso $w'[i] > w[i]$, é preciso verificar se o menor sufixo (lexicograficamente) em $w'_j[1, \dots]$ é maior do que $w'_j[0, \dots]$. Pode-se obter o menor sufixo de $w'_j[1, \dots]$ por meio da fatoração Lyndon de $w'_j[1, \dots]$, uma vez que o último fator em $CFL(w'_j[1, \dots])$ corresponde ao menor sufixo de $w'_j[1, \dots]$ (Propriedade 2.2.3). Esse passo leva tempo $O(n_j)$ utilizando o algoritmo de Duval (Seção 3.1). Em seguida, compara-se o menor sufixo de $w'_j[1, \dots]$ com $w'_j[0, \dots]$ em tempo $O(n_j)$, se ele for maior a propriedade (I) é válida.

□

Lema 4.1.2. *Considerando que o símbolo modificado está no início do fator w_j de $CFL(w)$, é possível verificar a propriedade (II) em tempo $O(n_j)$, com $|w_j| = n_j$.*

Demonstração. Caso $w'[i] < w[i]$, então $w_{j-1} > w'_j$, uma vez que anteriormente $w_{j-1} \geq w_j$. Em seguida, é preciso verificar se $w'_j \geq w_{j+1}$. Se $w'_j[0] > w_{j+1}[0]$, então a propriedade (II) vale. Se $w'_j[0] < w_{j+1}[0]$, então a propriedade (II) **não** vale. Se $w'_j[0] = w_{j+1}[0]$,

é necessário comparar o restante das duas palavras $w'_j[1, \dots]$ e $w_{j+1}[1, \dots]$. Para o caso onde $w'_j[1, \dots] \geq w_{j+1}[1, \dots]$, então a propriedade (II) vale. Caso contrário, **não** vale.

De forma simétrica, quando $w'[i] > w[i]$, o fator $w'_j > w_{j+1}$, uma vez que anteriormente $w_j \geq w_{j+1}$. Em seguida, é preciso verificar se $w_{j-1} \geq w'_j$. Se $w_{j-1}[0] > w'_j[0]$, então a propriedade (II) vale. Se $w'_j[0] > w_{j-1}[0]$, então a propriedade (II) **não** vale. Se $w'_j[0] = w_{j-1}[0]$, é necessário comparar o restante das duas palavras $w'_j[1, \dots]$ e $w_{j-1}[1, \dots]$. Para o caso onde $w_{j-1}[1, \dots] \geq w'_j[1, \dots]$, então a propriedade (II) vale. Caso contrário, **não** vale.

O custo total para verificar (II) é $O(n_j)$, uma vez que é necessário apenas comparar w'_j com os dois fatores adjacentes w_{j-1} e w_{j+1} .

□

Agora, será apresentado como atualizar $CFL(w)$ para obter $CFL(w')$ em tempo $O(n)$ no pior caso. Para isso, serão considerados os quatro possíveis casos referentes às propriedades (I) e (II).

4.1.1.1 Caso 1.

Lema 4.1.3. *Se (I) e (II) são satisfeitas, então $w'_j = w'_j[0]w_j[1, \dots]$ é um fator Lyndon, e os fatores Lyndon que antecedem e sucedem w'_j são os mesmos de $CFL(w)$, isto é,*

$$CFL(w') = w_0 \cdots w_{j-1} \cdot w'_j \cdot w_{j+1} \cdots w_{k-1}$$

A prova do lema acima é trivial, pois as duas propriedades são satisfeitas.

Exemplo 4.2. *Considere a palavra $w = deedeccdbdbdbd\mathbf{ad}adaaadaea$, e sua fatoração Lyndon:*

$$CFL(w) = dee \cdot de \cdot cd \cdot cd \cdot bdd \cdot bd \cdot bd \cdot \underbrace{\mathbf{ad}}_{w_7} \cdot ad \cdot aaadae \cdot a$$

Agora, considere que $w[16]$, que corresponde à posição inicial do fator $w_7 = ad$, seja alterado para **b**. Como $w'_7 = bd$ é um fator Lyndon e $w_6 \geq w'_7$ e $w'_7 \geq w_8$, então,

$$CFL(w') = dee \cdot de \cdot cd \cdot cd \cdot bdd \cdot bd \cdot bd \cdot \underbrace{\mathbf{bd}}_{w_7} \cdot ad \cdot aaadae \cdot a$$

Lema 4.1.4. *O custo para calcular $CFL(w')$ a partir de $CFL(w)$ quando (I) e (II) são satisfeitas é de $O(n_j)$.*

Demonstração. É possível verificar (I) e (II) em tempo $O(n_j)$ (Lemas 4.1.1 e 4.1.2). $\square$

4.1.1.2 Caso 2.

Lema 4.1.5. *Se (I) é satisfeita e (II) **não** é satisfeita, então $w'_j = w'_j[0]w_j[1, \dots]$ é um fator Lyndon, e apenas um dos seguintes casos pode acontecer, ou o fator $w_{j-1} \not\leq w'_j$ ou o fator $w'_j \not\leq w_{j+1}$.*

Caso $w_{j-1} \not\leq w'_j$, então,

$$CFL(w') = w_0 \cdots w_a \cdot \underbrace{w_{a+1} \cdots w_{j-1} w'_j}_{\text{novο fator}} \cdot w_{j+1} \cdots w_{k-1}$$

em que w_a é o primeiro fator à esquerda de w_{j-1} que é maior (lexicograficamente) do que a concatenação dos fatores $w_{a+1} \cdots w_{j-1} w'_j$. Os fatores que antecedem w_{a+1} e que sucedem w_j em $CFL(w)$ são os mesmos na nova fatoração.

Caso $w'_j \not\leq w_{j+1}$, então,

$$CFL(w') = w_0 \cdots w_{j-1} \cdot \underbrace{w'_j w_{j+1} \cdots w_{b-1}}_{\text{novο fator}} \cdot w_b \cdots w_{k-1}$$

em que w_b é o primeiro fator à direita de w'_j que é menor (lexicograficamente) do que a concatenação dos fatores $w'_j \cdots w_{b-1}$. Os fatores que sucedem w_{b-1} e que antecedem w_j em $CFL(w)$ são os mesmos na nova fatoração.

Demonstração. Por definição $w'_j = w'_j[0]w_j[1, \dots]$ é um fator Lyndon uma vez que (I) é satisfeita.

Assim será provado que só pode ocorrer uma das duas opções apresentadas no lema. Primeiro, observe que apenas um dos fatores adjacentes ao novo fator w'_j pode não respeitar a propriedade (II), ou seja, $w_{j-1} \not\leq w'_j$ e $w'_j \not\leq w_{j+1}$ não podem ocorrer simultaneamente. Quando $w_{j-1} \not\leq w'_j$ tem-se que $w'[i] > w[i]$, nesse caso $w'_j > w_j \geq w_{j+1}$, portanto, $w'_j \geq w_{j+1}$. De forma simétrica, quando $w'_j \not\leq w_{j+1}$, tem-se $w'[i] < w[i]$ e $w_{j-1} \geq w_j > w'_j$, portanto $w_{j-1} \geq w'_j$.

Considerando o caso em que $w_{j-1} \not\leq w'_j$, isto é, $w_{j-1} < w'_j$. Uma vez que w'_j é Lyndon, a concatenação $w_{j-1}w'_j$ também é um fator Lyndon (Proposição 2.2.1). Em seguida, se o fator Lyndon $w_{j-2} < w_{j-1}w'_j$, então a concatenação $w_{j-2}w_{j-1}w'_j$ também é um fator Lyndon. O mesmo vale para os fatores $w_{j-\ell}$ enquanto a concatenação com

$w_{j-\ell} \dots w_{j-2} w_{j-1} w'_j$ for maior do que o fator $w_{j-\ell-1}$, com $1 < \ell < j$. Suponha que isso aconteça até o fator w_{a+1} , portanto, $j - \ell = a + 1$. Dessa forma, $w_{a+1} \dots w_{j-2} w_{j-1} w'_j$ é um fator Lyndon, com $w_a > w_{a+1} \dots w'_j$. Além disso, por definição $w_{a+1} \geq w_{j+1}$, logo $w_{a+1} \dots w'_j > w_{j+1}$.

O caso em que $w'_j \not\geq w_{j+1}$ é simétrico ao anterior.

□

Exemplo 4.3. Considere a palavra $w = deede c d c d \mathbf{b} d d b d b d a d a d a a a d a e a$, e sua fatoração Lyndon:

$$CFL(w) = dee \cdot de \cdot cd \cdot cd \cdot \underbrace{bdd}_{w_4} \cdot bd \cdot bd \cdot ad \cdot ad \cdot aaadae \cdot a$$

Agora, considere que $w[9]$, que corresponde à posição inicial do fator $w_4 = bdd$, seja alterado para c . Primeiro, observe que $w'_4 = cdd$ é um fator Lyndon. Além disso, $w'_4 \geq w_5$ uma vez que $w'[9] > w[9]$. Então, tem-se que $w_3 \not\geq w'_4$, isto é, $cd \not\geq cdd$. Dessa forma, $w_3 w'_4 = cdcdd$ será considerado como um novo fator Lyndon. Em seguida, ao verificar que $w_2 = cd < cdcdd$, obtém-se $w_2 w_3 w'_4 = cdcdcd$ como um novo fator Lyndon. Por fim, é encontrado o fator $w_1 = de$, o qual é maior do que $cdcdcd$. Nesse momento, o processo de concatenação é interrompido e a nova fatoração para w' será igual à

$$CFL(w') = dee \cdot de \cdot \underbrace{cdcdcd}_{\text{novo fator}} \cdot bd \cdot bd \cdot ad \cdot ad \cdot aaadae \cdot a$$

Lema 4.1.6. O custo para calcular $CFL(w')$ a partir de $CFL(w)$ quando (I) é satisfeita e (II) **não** é satisfeita é de $O(n)$.

Demonstração. É possível verificar (I) e (II) em tempo $O(n_j)$ (Lemas 4.1.1 e 4.1.2).

Caso $w'[i] > w[i]$, será comparado w'_j com w_{j-1} . Essa comparação pode ser feita em tempo $O(n_j)$ comparando os símbolos dos dois fatores. Se $w_{j-1} < w'_j$, compara-se w_{j-2} com $w_{j-1} w'_j$, e assim por diante, até encontrar $w_a > w_{a+1} \dots w_{j-1} w'_j$, ou chegar no início de $CFL(w)$. A comparação de cada fator w_{j-i} com a concatenação $w_{j-i+1} \dots w_{j-1} w'_j$ é feita em tempo proporcional ao tamanho de w_{j-i} , portanto, o custo total até encontrar w_a é proporcional à soma de todos os fatores, isto é, $O(n_a + n_{a+1} + \dots + n_{j-1} + n_j) = O(n)$.

Caso $w'[i] < w[i]$, o custo é o mesmo, com as comparações feitas apenas para os sucessores de w_j em $CFL(w)$.

□

4.1.1.3 Caso 3.

Lema 4.1.7. *Se (I) não é satisfeita e (II) é satisfeita, então $w'_j = w'_j[0]w_j[1, \dots]$ não é um fator Lyndon.*

Seja $CFL(w'_j) = w'_{j_0} \cdot w'_{j_1} \cdots w'_{j_{q-1}}$, cujo o fator $w_{j-1} \geq w'_{j_0}$ e o fator $w'_{j_{q-1}} \geq w_{j+1}$, portanto,

$$CFL(w') = w_0 \cdots w_{j-1} \cdot \underbrace{w'_{j_0} \cdot w'_{j_1} \cdots w'_{j_{q-1}}}_{CFL(w'_j)} \cdot w_{j+1} \cdots w_{k-1}$$

Demonstração. Por definição $w'_j = w'_j[0]w_j[1, \dots]$ não é Lyndon uma vez que (I) não é satisfeita. Logo, $w'[i] > w[i]$.

Note que o fator w_{j-1} será maior ou igual ao primeiro fator w'_{j_0} de $CFL(w'_j)$, uma vez que pela propriedade (II), tem-se $w_{j-1} \geq w'_j$ e como w'_{j_0} é um prefixo de w'_j , logo $w_{j-1} \geq w'_{j_0}$.

Em contrapartida, o fator $w'_{j_{q-1}}$ de $CFL(w'_j)$ será maior ou igual fator w_{j+1} , uma vez que $w'_{j_{q-1}} > w_j \geq w_{j+1}$, e considerando que a alteração em $w[i]$ é feita no início do fator w'_j , tem-se $w'_{j_{q-1}} > w_{j+1}$.

□

Exemplo 4.4. *Considere a palavra $w = deede\mathbf{cd}cd\mathbf{b}dd\mathbf{b}dbdadadaaadaea$, e sua fatoração Lyndon:*

$$CFL(w) = dee \cdot de \cdot \underbrace{\mathbf{cd}}_{w_2} \cdot cd \cdot bdd \cdot bd \cdot bd \cdot ad \cdot ad \cdot aaadae \cdot a$$

Agora, considere que $w[5]$, que corresponde à posição do símbolo inicial do fator $w_2 = cd$, seja alterado para $\mathbf{d}$. Primeiro, observe que $w'_2 = dd$ não é um fator Lyndon e $CFL(dd) = d \cdot d$. Além disso, $w_1 \geq w'_2$ e $w'_2 \geq w_3$.

Portanto,

$$CFL(w') = dee \cdot de \cdot \underbrace{\mathbf{d \cdot d}}_{CFL(w'_2)} \cdot cd \cdot bdd \cdot bd \cdot bd \cdot ad \cdot ad \cdot aaadae \cdot a$$

Lema 4.1.8. *O custo para calcular $CFL(w')$ a partir de $CFL(w)$ quando (I) **não** é satisfeita e (II) é satisfeita é de $O(n_j)$.*

Demonstração. É possível verificar (I) e (II) em tempo $O(n_j)$ (Lemas 4.1.1 e 4.1.2).

Em seguida, $CFL(w'_j)$ pode ser calculada em tempo $O(n_j)$ utilizando o algoritmo de Duval (Seção 3.1). □

4.1.1.4 Caso 4.

Lema 4.1.9. *Dada alteração no início de um fator w_j de $CFL(w)$ e as propriedades (I) e (II) não forem satisfeitas, então apenas os fatores à esquerda de w_j podem sofrer alterações. Portanto, os fatores à direita de w_j permanecem os mesmos.*

Demonstração. Por definição $w'_j = w'_j[0]w_j[1, \dots]$ é maior que algum sufixo próprio de w'_j uma vez que (I) não é satisfeita. Então, $w'[i] > w[i]$.

Note que $w'_j \geq w_{j+1}$ uma vez que $w'_j \geq w_j \geq w_{j+1}$. Portanto, apenas o caso $w_{j-1} \not\geq w'_j$ pode acontecer quando a propriedade (II) não for satisfeita. □

Lema 4.1.10. *Se (I) **não** é satisfeita e (II) **não** é satisfeita, então $w'_j = w'_j[0]w_j[1, \dots]$ não é um fator Lyndon. Além disso, apenas o caso em que $w_{j-1} \not\geq w'_{j_0}$ pode acontecer e a fatoração de w' é dada por:*

$$CFL(w') = w_0 \cdots w_a \cdot \underbrace{w_{a+1} \cdots w_{j-1} w'_{j_0} \cdots w'_{j_{b-1}}}_{\text{novo fator}} \cdot \underbrace{w'_{j_b} \cdots w'_{j_{q-1}}}_{\text{novo fator}} \cdot w_{j+1} \cdots w_{k-1}$$

em que

$$CFL(w'_j) = w'_{j_0} \cdots w'_{j_{b-1}} \cdots w'_{j_b} \cdots w'_{j_{q-1}}$$

cujo o fator Lyndon w'_{j_b} é o primeiro em $CFL(w'_j)$ que não é igual a w'_{j_0} . Se tal fator não existir em $CFL(w'_j)$, então $j_{b-1} = j_{q-1}$, isto é, $w'_{j_{b-1}}$ é o último fator de $CFL(w'_j)$. O fator w_a é o primeiro fator Lyndon de $CFL(w)$ à esquerda de w_{j-1} que é maior ou igual (lexicograficamente) do que a concatenação dos fatores $w_{a+1} \cdots w_{j-1} w'_{j_0} \cdots w'_{j_{b-1}}$. Os fatores que antecedem w_{a+1} e que sucedem $w'_{j_{q-1}}$ em $CFL(w)$ são os mesmos na nova fatoração.

Demonstração. Pelo Lema 4.1.9, apenas o caso $w_{j-1} \not\geq w'_j$ pode acontecer.

Seja $CFL(w'_j) = w'_{j_0} \cdots w'_{j_{b-1}} \cdots w'_{j_b} \cdots w'_{j_{q-1}}$ a nova fatoração de w'_j , tem-se $w_{j-1} < w'_{j_0}$, pois $w_{j-1} \not\geq w'_j$, e como os dois fatores são Lyndon, $w_{j-1}w'_{j_0}$ é um fator Lyndon (Proposição 2.2.1). Agora, se $w'_{j_0} = w'_{j_1}$, então $w_{j-1}w'_{j_0} < w'_{j_1}$ e $w_{j-1}w'_{j_0}w'_{j_1}$ também é um fator Lyndon. O mesmo acontece para todos os fatores Lyndon iguais que sucedem w'_{j_1} em $CFL(w'_j)$, isto é, se $w'_{j_0} = w'_{j_1} = w'_{j_2} = \cdots = w'_{j_{b-1}}$, então $w_{j-1}w'_{j_0} \cdots w'_{j_{b-2}} < w'_{j_{b-1}}$ e $w_{j-1}w'_{j_0} \cdots w'_{j_{b-1}}$ também é um fator Lyndon.

Em seguida, semelhante ao **Caso 2**, é preciso comparar os fatores que antecedem w_{j-1} com a concatenação dos fatores que o seguem até $w'_{j_{b-1}}$ enquanto esses fatores forem menores (lexicograficamente) do que a concatenação. Suponha que isso aconteça até o fator w_{a+1} . Dessa forma, $w_{a+1} \cdots w_{j-2}w_{j-1}w'_{j_0} \cdots w'_{j_{b-1}}$ é um fator Lyndon, com $w_a \geq w_{a+1} \cdots w_{j-1}w'_{j_0} \cdots w'_{j_{b-1}}$ e, além disso, $w_{a+1} \cdots w'_{j_0} \cdots w'_{j_{b-1}} > w'_{j_b}$ já que $w_{a+1} \geq w_j$ e o sufixo que começa em w_{a+1} é maior do que o sufixo que começa em w'_{j_b} , uma vez que a atualização foi feita com $w'[i] > w[i]$.

□

Exemplo 4.5. Considere a palavra $w = deedecdcdbddbdbdadadaaadaea$, e sua fatoração Lyndon:

$$CFL(w) = dee \cdot de \cdot cd \cdot cd \cdot \underbrace{bdd \cdot bd \cdot bd \cdot ad \cdot ad}_{w_4} \cdot aaadae \cdot a$$

Agora, considere que $w[9]$, que corresponde à posição do símbolo inicial do fator $w_4 = bdd$, seja alterado para **d**. Primeiro, observe que $w'_4 = ddd$ não é um fator Lyndon. Então, calcula-se $CFL(w'_4) = d \cdot d \cdot d$, portanto, $w'_{j_0} = d$ é um fator Lyndon, e como $w_{j-1} < w'_{j_0}$, a concatenação $w_{j-1}w'_{j_0} = cdd$ é um fator Lyndon (Proposição 2.2.1). Esse fator Lyndon, por sua vez, é menor do que $w'_{j_1} = d$, então $w_{j-1}w'_{j_0}w'_{j_1} = cddd$ também é um fator Lyndon. O mesmo acontece com o fator Lyndon $w'_{j_2} = d$, isto é $w_{j-1}w'_{j_0}w'_{j_1} < d$, portanto a concatenação $w_{j-1}w'_{j_0}w'_{j_1}w'_{j_2} = cdddd$ também é um fator Lyndon. Agora, é preciso verificar se o fator Lyndon $w_{j-2} = cd$ é menor do que $cdddd$, o qual nesse caso $w_{j-2}w_{j-1}w'_{j_0}w'_{j_1} = cdcdddd$ também é um fator Lyndon. Por fim, verifica-se que $w_{j-3} = de$ é maior do que $cdcd$, portanto a atualização de $CFL(w)$ é interrompida e obtendo:

$$CFL(w') = dee \cdot de \cdot \underbrace{cdcd}_{\text{novo fator}} \cdot bdd \cdot bd \cdot ad \cdot ad \cdot aaadae \cdot a$$

Lema 4.1.11. *O custo para calcular $CFL(w')$ a partir de $CFL(w)$ quando (I) e (II) **não** são satisfeitas é de $O(n)$.*

A prova é igual à do Lema 4.1.6, considerando apenas o caso em que os fatores à esquerda de w_j são comparados com os fatores em $CFL(w')$.

4.1.2 Alteração em qualquer posição de um fator

Agora, considere que a posição do símbolo $w[i]$ alterado corresponde a uma posição $p > 0$ de um fator Lyndon w_j de $CFL(w)$, portanto a subpalavra w'_j pode ser escrita como $w_j[0, p-1]w'_j[p]w_j[p+1, ..]$.

Inicialmente, será apresentado como verificar as propriedades (I) e (II) após a alteração de $w[i]$ em tempo $O(n_j)$, com $|w_j| = n_j$.

Lema 4.1.12. *Considerando que o símbolo modificado está na posição $p > 0$ do fator w_j de $CFL(w)$, é possível verificar a propriedade (I) em tempo $O(n_j)$, com $|w_j| = n_j$.*

Demonstração. Primeiro, é preciso verificar se w'_j continua sendo menor que seus sufixos próprios. Para isso, é realizada a fatoração Lyndon $CFL(w'_j)$. Esse passo leva tempo $O(n_j)$ utilizando o algoritmo de Duval (Seção 3.1).

□

Lema 4.1.13. *Considerando que o símbolo modificado está na posição $p > 0$ do fator w_j de $CFL(w)$, é possível verificar a propriedade (II) em tempo $O(n_j)$, com $|w_j| = n_j$.*

Demonstração. Caso $w'[i] < w[i]$, então $w_{j-1} > w'_j$, uma vez que anteriormente $w_{j-1} \geq w_j$. Em seguida, é preciso verificar se $w'_j \geq w_{j+1}$. Se $w'_j[p] > w_{j+1}[p]$, então a propriedade (II) vale. Caso $w'_j[p] \leq w_{j+1}[p]$, é preciso comparar as duas palavras w'_j e w_{j+1} . Para o caso $w'_j \geq w_{j+1}$, então a propriedade (II) vale. Caso contrário, **não** vale.

De forma simétrica, quando $w'[i] > w[i]$, o fator $w'_j > w_{j+1}$, uma vez que anteriormente $w_j \geq w_{j+1}$. Em seguida, é preciso verificar se $w_{j-1} \geq w'_j$. Se $w'_j[p] < w_{j-1}[p]$, então a propriedade (II) vale. Caso $w'_j[p] \geq w_{j-1}[p]$, é preciso comparar as duas palavras w'_j e w_{j+1} . Para o caso $w_{j-1} \geq w'_j$, então a propriedade (II) vale. Caso contrário, **não** vale.

O custo total para verificar (II) é $O(n_j)$, uma vez que é necessário apenas comparar w'_j com os dois fatores adjacentes w_{j-1} e w_{j+1} .

□

Agora, será apresentado como atualizar $CFL(w)$ para obter $CFL(w')$ em tempo $O(k + n)$, onde k é o número de fatores em $CFL(w)$. Serão considerados os quatro possíveis casos referente às propriedades (I) e (II).

4.1.2.1 Caso 5.

Lema 4.1.14. *Se (I) e (II) são satisfeitas, então $w'_j = w_j[0, p - 1]w'[p]w_j[p + 1, ..]$ é um fator Lyndon, e os fatores Lyndon que antecedem e sucedem w'_j são os mesmos de $CFL(w)$, isto é,*

$$CFL(w') = w_0 \cdots w_{j-1} \cdot w'_j \cdot w_{j+1} \cdots w_{k-1}$$

A prova do lema acima é trivial.

Exemplo 4.6. *Considere a palavra $w = deedeccdcdbddbdbdadadaaadaea$, e sua fatoração Lyndon:*

$$CFL(w) = dee \cdot de \cdot cd \cdot cd \cdot \underbrace{bdd}_{w_4} \cdot bd \cdot bd \cdot ad \cdot ad \cdot aaadae \cdot a$$

Agora, considere que $w[11]$, que corresponde à posição $p = 3$ do fator $w_4 = bdd$, seja alterado para **e**. Como $w'_4 = bde$ é um fator Lyndon e $w_3 \geq w'_4$ e $w'_4 \geq w_5$, então,

$$CFL(w) = dee \cdot de \cdot cd \cdot cd \cdot \underbrace{bde}_{w_4} \cdot bd \cdot bd \cdot ad \cdot ad \cdot aaadae \cdot a$$

Lema 4.1.15. *O custo para calcular $CFL(w')$ a partir de $CFL(w)$ quando (I) e (II) são satisfeitas é de $O(n_j)$.*

Demonstração. É possível verificar (I) e (II) em tempo $O(n_j)$ (Lemas 4.1.1 e 4.1.2). □

4.1.2.2 Caso 6.

Lema 4.1.16. *Se (I) é satisfeita e (II) não é satisfeita, então o fator $w'_j = w_j[0, p - 1]w'[p]w_j[p + 1, ..]$ é um fator Lyndon, e apenas um dos seguintes casos pode acontecer, ou o fator $w_{j-1} \not\geq w'_j$ ou o fator $w'_j \not\geq w_{j+1}$.*

Caso $w_{j-1} \not\preceq w'_j$, então,

$$CFL(w') = w_0 \cdots w_a \cdot \underbrace{w_{a+1} \cdots w_{j-1} w'_j}_{\text{novo fator}} \cdot w_{j+1} \cdots w_{k-1}$$

em que w_a é o primeiro fator à esquerda de w_{j-1} que é maior (lexicograficamente) do que a concatenação dos fatores $w_{a+1} \cdots w_{j-1} w'_j$. Os fatores que antecedem w_{a+1} e que sucedem w_j em $CFL(w)$ são os mesmos na nova fatoração.

Caso $w'_j \not\preceq w_{j+1}$, então,

$$CFL(w') = w_0 \cdots w_{j-1} \cdot \underbrace{w'_j w_{j+1} \cdots w_{b-1}}_{\text{novo fator}} \cdot w_b \cdots w_{k-1}$$

em que w_b é o primeiro fator à direita de w'_j que é menor (lexicograficamente) do que a concatenação dos fatores $w'_j \cdots w_{b-1}$. Os fatores que sucedem w_{b-1} e que antecedem w_j em $CFL(w)$ são os mesmos na nova fatoração.

A prova do lema acima é a mesma do Lema 4.1.5.

Exemplo 4.7. Considere a palavra $w = \text{deedecdcdbddbdb}\mathbf{d}\text{adadaaadaea}$, e sua fatoração Lyndon:

$$CFL(w) = \text{dee} \cdot \text{de} \cdot \text{cd} \cdot \text{cd} \cdot \text{bdd} \cdot \text{bd} \cdot \underbrace{\mathbf{bd}}_{w_6} \cdot \text{ad} \cdot \text{ad} \cdot \text{aaadae} \cdot \text{a}$$

Agora, considere que $w[16]$, que corresponde à posição $p = 1$ do fator $w_6 = \mathbf{bd}$, seja alterado para $\mathbf{e}$. Primeiro, observe que $w'_6 = \mathbf{be}$ é um fator Lyndon. Além disso, $w'_6 \geq w_7$ e $w_5 \not\preceq w'_6$. Dessa forma, $w_5 w'_6 = \mathbf{bdbe}$ será um novo fator Lyndon (Proposição 2.2.1). Por fim, é encontrado o fator $w_4 = \mathbf{bdd}$ que é maior do que $\mathbf{bdbe}$. Nesse momento, o processo de concatenação é interrompido e a nova fatoração para w' é igual à

$$CFL(w) = \text{dee} \cdot \text{de} \cdot \text{cd} \cdot \text{cd} \cdot \text{bdd} \cdot \underbrace{\mathbf{bdbe}}_{\text{novo fator}} \cdot \text{ad} \cdot \text{ad} \cdot \text{aaadae} \cdot \text{a}$$

Lema 4.1.17. O custo para calcular $CFL(w')$ a partir de $CFL(w)$ quando (I) é satisfeita e (II) **não** é satisfeita é de $O(n)$.

A prova do lema acima é a mesma do Lema 4.1.6.

4.1.2.3 Caso 7.

Lema 4.1.18. *Se (I) não é satisfeita e (II) é satisfeita, então $w'_j = w_j[0, p - 1]w'[p]w_j[p + 1, ..]$ não é um fator Lyndon.*

Seja $CFL(w'_j) = w'_{j_0} \cdot w'_{j_1} \cdots w'_{j_{q-1}}$, tem-se que $w_{j-1} \geq w'_{j_0}$ e, portanto,

$$CFL(w') = w_0 \cdots w_{j-1} \cdot \underbrace{w'_{j_0} \cdots w'_{j_{r-1}}}_{\text{novo fator}} \cdot \underbrace{w'_{j_r} \cdots w'_{j_{q-1}} w_{j+1} \cdots w_{b-1} \cdot w_b \cdots w_{k-1}}_{\text{novo fator}}$$

em que w_b é o primeiro fator à direita de w'_j que é menor (lexicograficamente) do que a concatenação dos fatores $w'_{j_r} \cdots w_{b-1}$, e $w'_{j_{r-1}}$ é o primeiro fator que é maior (lexicograficamente) do que a concatenação dos fatores $w'_{j_r} \cdots w'_{j_{q-1}} w_{j+1}$. Além disso, os fatores que sucedem w_{b-1} e que antecedem w_j em $CFL(w)$ são os mesmos na nova fatoração.

Demonstração. Por definição $w'_j = w_j[0, p - 1]w'[p]w_j[p + 1, ..]$ é maior que algum sufixo próprio de w'_j uma vez que (I) não é satisfeita.

Note que o fator w_{j-1} será maior ou igual ao primeiro fator w'_{j_0} de $CFL(w'_j)$, uma vez que pela propriedade (II), tem-se $w_{j-1} \geq w'_j$ e como w'_{j_0} é um prefixo de w'_j , logo $w_{j-1} > w'_{j_0}$.

Observe que o último fator $w'_{j_{q-1}}$ de $CFL(w'_j)$ pode não ser maior ou igual fator w_{j+1} , uma vez que a alteração em $w[i]$ pode ter sido feita em $w'_{j_{q-1}}$. Nesse caso, se $w'_{j_{q-1}} < w_{j+1}$, então $w'_{j_{q-1}} w_{j+1}$ é um fator Lyndon.

Agora, se $w'_{j_{q-2}} \not\geq w'_{j_{q-1}} w_{j+1}$, então a concatenação $w'_{j_{q-2}} w'_{j_{q-1}} w_{j+1}$ também é um fator Lyndon. Assim, essa comparação e concatenação é feita enquanto os fatores que antecedem $w'_{j_{q-1}}$ em $CFL(w'_j)$ forem menores (lexicograficamente) do que a concatenação. Observe que, nesse caso, a única possibilidade de isso ocorrer é se esses fatores forem iguais, pois $w'_{j_{q-2}} \geq w'_{j_{q-1}}$ e para $w'_{j_{q-2}} < w'_{j_{q-1}} w_{j+1}$ é preciso que $w'_{j_{q-2}} = w'_{j_{q-1}}$ (o mesmo vale para os fatores que antecedem $w'_{j_{q-2}}$). Suponha que isso aconteça até o fator $w'_{j_{r-1}}$, com $0 < r < q$.

Em seguida, semelhante ao **Caso 2**, é necessário comparar os fatores que sucedem w_{j+1} com a concatenação dos fatores que o seguem até w'_{j_r} enquanto eles forem maiores (lexicograficamente) do que a concatenação. Suponha que isso aconteça até o fator w_{b-1} . Dessa forma, $w'_{j_r} \cdots w'_{j_{q-1}} w_{j+1} \cdots w_{b-1}$ é um fator Lyndon, com $w'_{j_r} \cdots w'_{j_{q-1}} w_{j+1} \cdots w_{b-1} > w_b$. Além disso, $w_{j-1} \geq w_{b-1}$, logo $w_{j-1} \geq \cdots \geq w'_{j_{r-1}} > w'_{j_r} \cdots w'_{j_{q-1}} w_{j+1} \cdots w_{b-1} \geq w_b \geq \cdots \geq w_{k-1}$.

□

Exemplo 4.8. Considere a palavra $w = deedeccdbdd**b**dbdadadaaadaea$, e sua fatoração Lyndon:

$$CFL(w) = dee \cdot de \cdot cd \cdot cd \cdot \underbrace{bdd}_{w_4} \cdot \underbrace{bd}_{w_5} \cdot \underbrace{bd}_{w_6} \cdot \underbrace{ad}_{w_7} \cdot ad \cdot aaadae \cdot a$$

Agora, considere que $w[11]$, que corresponde à posição $p = 2$ do fator $w_4 = bdd$, seja alterado para **b**. Primeiro, observe que $w'_4 = bdb$ não é um fator Lyndon. Tem-se $CFL(w'_4) = bd \cdot b$. Além disso, $w_3 \geq w'_4$ e $w'_4 \geq w_5$. O último fator de $CFL(w'_4)$ é menor do que w_5 , portanto, bbd é um fator Lyndon. O mesmo acontece com o fator $w_6 = bd$, isto é $bbd < bd$, então, a concatenação $bbdbd$ também é um fator Lyndon. Por fim, é verificado que $w_7 = ad$ é menor do que $bbdbd$, portanto a atualização de $CFL(w)$ é interrompida e:

$$CFL(w) = dee \cdot de \cdot cd \cdot cd \cdot bd \cdot \underbrace{bd}_{w'_{j_0}} \cdot \underbrace{bbdbd}_{\text{novo fator}} \cdot ad \cdot ad \cdot aaadae \cdot a$$

Lema 4.1.19. O custo para calcular $CFL(w')$ a partir de $CFL(w)$ quando (I) **não** é satisfeita e (II) é satisfeita é $O(n)$.

Demonstração. É possível verificar (I) e (II) em tempo $O(n_j)$ (Lemas 4.1.12 e 4.1.13). Como (I) não é satisfeita, é preciso fazer o $CFL(w'_j) = w'_{j_0} \cdots w'_{j_{q-1}}$ cujo custo é dado por $O(n_j)$. Caso $w'[i] > w[i]$, será comparado $w'_{j_{q-1}}$ com w_{j+1} . Essa comparação pode ser feita em tempo $O(n_j)$ comparando os símbolos dos dois fatores. Se $w_{j+1} > w'_{j_{q-1}}$, compara-se w_{j+2} com $w'_{j_{q-1}} w_{j+1}$, e assim por diante, até encontrar $w_b < w'_{j_{q-1}} w_{j+1} \cdots w_{b-1}$, ou chegar no fim de $CFL(w)$. A comparação de cada fator w_{j+i} com a concatenação $w'_{j_{q-1}} w_{j+1} \cdots w_{j+i-1}$ é feita em tempo proporcional ao tamanho de w_{j+i} , portanto, o custo total até encontrar w_b é proporcional à soma de todos os fatores, isto é, $O(n'_{j_{q-1}} + n_j + n_{j+1} + \cdots + n_{b-1} + n_b) = O(n)$. □

4.1.2.4 Caso 8.

Lema 4.1.20. Se (I) **não** é satisfeita e (II) **não** é satisfeita, então $w'_j = w_j[0, p-1]w'[p]w_j[p+1, \dots]$ não é um fator Lyndon, e apenas um dos seguintes casos pode acontecer: ou o fator $w_{j-1} \not\leq w'_j$ ou o fator $w'_j \not\leq w_{j+1}$. Seja $CFL(w'_j) = w'_{j_0} \cdot w'_{j_1} \cdots w'_{j_{q-1}}$.

Caso $w_{j-1} \not\leq w'_j$, a fatoração é dada por

$$CFL(w') = w_0 \cdots w_a \cdot \underbrace{w_{a+1} \cdots w_{j-1} w'_{j_0} \cdots w'_{j_{r-1}}}_{\text{novο fator}} \cdot \underbrace{w'_{j_r} \cdots w'_{q-1}}_{\text{novο fator}} \cdot w_{j+1} \cdots w_{k-1}$$

em que w_a é o primeiro fator à esquerda de w_{j-1} que é maior (lexicograficamente) do que a concatenação dos fatores $w_{a+1} \cdots w_{j-1} w'_{j_{r-1}}$, e w'_{j_r} é o primeiro fator que é menor (lexicograficamente) do que o fator w_{j-1} . Os fatores que antecedem w_{a+1} e que sucedem w_j em $CFL(w)$ são os mesmos na nova fatoração.

Caso $w'_j \not\leq w_{j+1}$, a fatoração é dada por

$$CFL(w') = w_0 \cdots w_{j-1} \cdot \underbrace{w'_{j_0} \cdots w'_{j_{r-1}}}_{\text{novο fator}} \cdot \underbrace{w'_{j_r} \cdots w'_{q-1} w_{j+1} \cdots w_{b-1}}_{\text{novο fator}} \cdot w_b \cdots w_{k-1}$$

em que w_b é o primeiro fator à direita de w'_j que é menor (lexicograficamente) do que a concatenação dos fatores $w'_{j_r} \cdots w_{b-1}$, e $w'_{j_{r-1}}$ é o primeiro fator que é maior (lexicograficamente) do que o fator w_{j+1} . Os fatores que sucedem w_{b-1} e que antecedem w_j em $CFL(w)$ são os mesmos na nova fatoração.

Demonstração. Por definição $w'_j = w_j[0, p-1]w'[p]w_j[p+1, \dots]$ não é o menor sufixo em w'_j uma vez que (I) não é satisfeita. Como $p > 0$, pode-se ter $w'[i] > w[i]$ ou $w'[i] < w[i]$. Assim, apenas um dos dois fatores Lyndon adjacentes à w'_j pode não respeitar a propriedade (II). Quando $w_{j-1} \not\leq w'_j$ tem-se que $w'[i] > w[i]$, nesse caso $w'_j > w_j \geq w_{j+1}$, portanto, $w'_j \geq w_{j+1}$. De forma simétrica, quando $w'_j \not\leq w_{j+1}$, tem-se $w'[i] < w[i]$ e $w_{j-1} \geq w_j > w'_j$, portanto $w_{j-1} \geq w'_j$.

Suponha que $w'_j > w_j$, então apenas o caso $w_{j-1} \not\leq w'_j$ pode acontecer quando a propriedade (II) não for satisfeita.

Considerando, $CFL(w'_j) = w'_{j_0} \cdots w'_{j_{r-1}} \cdots w'_{j_r} \cdots w'_{j_{q-1}}$, tem-se que $w_{j-1} < w'_{j_0}$, pois $w_{j-1} \not\leq w'_j$, e como os dois fatores são Lyndon, $w_{j-1} w'_{j_0}$ é um fator Lyndon (Proposição 2.2.1). Agora, se $w_{j-1} w'_{j_0} \not\leq w'_{j_1}$, então a concatenação $w_{j-1} w'_{j_0} w'_{j_1}$ é um fator Lyndon. Note que para verificar essa desigualdade basta comparar os fatores w'_{j_r} com w_{j-1} , com $0 < r < q$, uma vez que os fatores concatenados à w_{j-1} são maiores ou iguais à w'_{j_r} . Assim, essa comparação e concatenação é feita enquanto os fatores que sucedem w'_{j_0} em $CFL(w'_j)$ forem maiores (lexicograficamente) do que w_{j-1} .

Em seguida, semelhante ao **Caso 2**, é necessário comparar os fatores que antecedem w_{j-1} com a concatenação dos fatores que o seguem até $w'_{j_{r-1}}$ enquanto eles forem

menores (lexicograficamente) do que a concatenação. Suponha que isso aconteça até o fator w_{a+1} . Dessa forma, $w_{a+1} \dots w_{j-2} w_{j-1} w'_{j_0} \dots w'_{j_{r-1}}$ é um fator Lyndon, com $w_a > w_{a+1} \dots w'_{j_{r-1}}$. Além disso, $w_{a+1} \geq w_{j+1}$, logo $w_{a+1} \dots w'_{j_0} \dots w'_{j_{r-1}} > w'_{j_r} \geq \dots \geq w_{j+1}$.

De forma simétrica, quando $w'_j < w_j$ apenas o caso $w'_j \not\geq w_{j+1}$ pode acontecer quando a propriedade (II) não for satisfeita, cuja demonstração é a mesma do caso 4.1.18.

□

Exemplo 4.9. Considere a palavra $w = deedeccdbdd**bd**dadadaaadaea$, e sua fatoração Lyndon:

$$CFL(w) = dee \cdot de \cdot cd \cdot cd \cdot bdd \cdot \underbrace{bd}_{w_5} \cdot \underbrace{bd}_{w_6} \cdot \underbrace{ad}_{w_7} \cdot \underbrace{ad}_{w_8} \cdot aaadae \cdot a$$

Agora, considere que $w[13]$, que corresponde à posição $p = 1$ do fator $w_5 = bd$, seja alterado para **a**. Primeiro, observe que $w'_5 = ba$ não é um fator Lyndon e $w'_j \not\geq w_{j+1}$.

Então, é calculado $CFL(w'_5) = b \cdot a$, portanto, $w'_{j_1} = a$ é um fator Lyndon, e como $w'_{j_1} < w_6$, a concatenação $w'_{j_1} w_6 = abd$ é um fator Lyndon (Proposição 2.2.1). Esse fator Lyndon, por sua vez, é menor do que $w_7 = ad$, então $w'_{j_1} w_6 w_7 = abdad$ também é um fator Lyndon. O mesmo acontece para $w_8 = ad$, portanto a concatenação $w'_{j_1} w_6 w_7 w_8 = abdabad$ também é um fator Lyndon. Por fim, é verificado que $w_9 = aaadae$ é menor do que $abdabad$, portanto a atualização de $CFL(w')$ é interrompida e:

$$CFL(w') = dee \cdot de \cdot cd \cdot cd \cdot bdd \cdot b \cdot \underbrace{abdabad}_{\text{novo fator}} \cdot aaadae \cdot a$$

Exemplo 4.10. Considere a palavra $w = deedeccdbddbdbdadada**a**adaea$, e sua fatoração Lyndon:

$$CFL(w) = dee \cdot de \cdot cd \cdot cd \cdot bdd \cdot bd \cdot bd \cdot \underbrace{ad}_{w_7} \cdot \underbrace{ad}_{w_8} \cdot \underbrace{aaadae}_{w_9} \cdot a$$

Agora, considere que $w[20]$, que corresponde à posição $p = 1$ do fator $w_9 = aaadae$, seja alterado para **a**eadae. Primeiro, observe que $w'_9 = aeadae$ não é um fator Lyndon e $w_{j-1} \not\geq w'_j$.

Então, é calculado $CFL(w'_9) = ae \cdot adae$ e como $w'_{j_0} > w_{j-1}$, a concatenação $w_8 w'_{j_0} = adae$ é um fator Lyndon (Proposição 2.2.1). Note que, o fator $w_7 = ad$ é menor que o novo fator obtido $w_8 w'_{j_0} = adae$, resultando em um fator Lyndon $w_7 w_8 w'_{j_0} = adadae$. A seguir, é verificado que o fator $w_6 = bd$ é maior do que $w_7 w_8 w'_{j_0} = adadae$, portanto $w_a = w_6$.

Em seguida, verifica-se que a concatenação $w_7 w_8 w'_{j_0} = adadae$ é menor do que o fator $w'_{j_2} = adae$, logo $w_7 w_8 w'_{j_0} w'_{j_2} = adadaeadae$ é um novo fator Lyndon. Por fim, $w_{10} = a$ é menor do que o novo fator, portanto, a atualização de $CFL(w')$ é finalizada com:

$$CFL(w') = dee \cdot de \cdot cd \cdot cd \cdot bdd \cdot bd \cdot \underbrace{adadaeadae}_{\text{novo fator}} \cdot a$$

Lema 4.1.21. O custo para calcular $CFL(w')$ a partir de $CFL(w)$ quando (I) e (II) não são satisfeitas é de $O(n)$.

A prova é igual à do Lema 4.1.6.

4.1.3 Considerações Finais

Seja w uma palavra de tamanho n , sua fatoração Lyndon $CFL(w)$, e $w[i]$ uma posição em um fator w_j de $CFL(w)$.

Teorema 4.1.22. É possível verificar se a fatoração Lyndon após uma operação de substituição em $w[i]$ é a mesma em tempo $O(n_j)$, em que n_j é o tamanho do fator em que o símbolo $w[i]$ foi alterado.

Demonstração. Quando uma alteração é feita em qualquer posição de um fator de $CFL(w)$, pelos Lemas 4.1.1, 4.1.2, 4.1.12 e 4.1.13 verifica-se as condições (I) e (II) em tempo $O(n_j)$. $\square$

Teorema 4.1.23. É possível obter a nova fatoração Lyndon $CFL(w')$ para a palavra modificada w' após uma operação de substituição em $w[i]$, a partir da fatoração $CFL(w)$, em tempo $O(n)$.

Demonstração. Para os casos em que $CFL(w') = CFL(w)$, pelos Lemas 4.1.4 e 4.1.15, o custo para obter $CFL(w')$ é $O(n_j)$, em que n_j é o tamanho do fator em que o símbolo $w[i]$ foi alterado. Além disso, quando a alteração é feita no primeiro símbolo de um

fator e a condição (II) é satisfeita, pelo Lema 4.1.8, o custo para obter $CFL(w')$ também é $O(n_j)$. Para todos os outros casos, pelos Lemas 4.1.6, 4.1.11, 4.1.17, 4.1.19 e 4.1.21, o custo para obter $CFL(w')$ a partir de $CFL(w)$ é $O(n)$. $\square$

Conclusão

Neste trabalho foi investigado o problema da fatoração Lyndon dinâmica de uma palavra w no cenário em que uma operação de substituição é permitida na edição de w , de forma que a fatoração da nova palavra resultante w' pode ser obtida a partir da fatoração de w .

Inicialmente, foi feito um estudo sobre palavras Lyndon, fatoração Lyndon e algoritmos relacionados. A partir desse estudo, surgiu a seguinte pergunta sobre o que aconteceria com a fatoração de w se um símbolo fosse substituído: *Será que é possível encontrar os fatores Lyndon da palavra alterada w' a partir da fatoração de w sem ter que reprocessar todo o texto novamente?* Essa pergunta foi respondida com o desenvolvimento de um novo algoritmo que divide em oito possíveis casos que podem acontecer com a fatoração de w após uma operação de substituição.

O algoritmo apresentado primeiro verifica se a fatoração Lyndon após uma operação de substituição em $w[i]$ é a mesma em tempo $O(n_j)$, em que n_j é o tamanho do fator em que o símbolo $w[i]$ foi alterado. Caso seja necessário atualizar $CFL(w)$ para obter a nova fatoração Lyndon $CFL(w')$ para a palavra modificada w' o custo computacional em cinco casos é $O(n)$.

Esse trabalho foi um estudo inicial a respeito da atualização de fatores Lyndon existindo várias oportunidades de melhorias em trabalhos futuros.

Como questão em aberto fica se é possível obter a fatoração Lyndon de w' a partir de $CFL(w)$ em tempo $o(n)$ utilizando estruturas de dados auxiliares. Este desafio abre caminho para investigações futuras, particularmente na exploração da eficiência

algorítmica através de um estudo experimental.

Referências

BANNAI, H. et al. The “runs” theorem. **SIAM Journal on Computing**, Society for Industrial Applied Mathematics (SIAM), v. 46, n. 5, p. 1501–1514, jan. 2017. ISSN 1095-7111. Disponível em: <<http://dx.doi.org/10.1137/15M1011032>>.

_____. Computing longest (common) lyndon subsequences. In: BAZGAN, C.; FERNAU, H. (Ed.). **Combinatorial Algorithms - 33rd International Workshop, IWOCA 2022, Trier, Germany, June 7-9, 2022, Proceedings**. Springer, 2022. (Lecture Notes in Computer Science, v. 13270), p. 128–142. Disponível em: <https://doi.org/10.1007/978-3-031-06678-8_10>.

_____. Indexing the bijective BWT. In: PISANTI, N.; PISSIS, S. P. (Ed.). **30th Annual Symposium on Combinatorial Pattern Matching, CPM 2019, June 18-20, 2019, Pisa, Italy**. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. (LIPIcs, v. 128), p. 17:1–17:14. Disponível em: <<https://doi.org/10.4230/LIPIcs.CPM.2019.17>>.

BONIZZONI, P. et al. Kfinger: Capturing overlaps between long reads by using lyndon fingerprints. In: ROJAS, I. et al. (Ed.). **Bioinformatics and Biomedical Engineering - 9th International Work-Conference, IWBBIO 2022, Maspalomas, Gran Canaria, Spain, June 27-30, 2022, Proceedings, Part II**. Springer, 2022. (Lecture Notes in Computer Science, v. 13347), p. 436–449. Disponível em: <https://doi.org/10.1007/978-3-031-07802-6_37>.

CHEN, K. T.; FOX, R. H.; LYNDON, R. C. Free differential calculus, IV. the quotient groups of the lower central series. **The Annals of Mathematics**, JSTOR, v. 68, n. 1, p. 81, jul. 1958. Disponível em: <<https://doi.org/10.2307/1970044>>.

CROCHEMORE, M.; RYTTER, W. **Jewels of stringology**. World Scientific, 2002. ISBN 978-981-02-4782-9. Disponível em: <<https://doi.org/10.1142/4838>>.

- DELGRANGE, O.; RIVALS, E. STAR: an algorithm to search for tandem approximate repeats. **Bioinform.**, v. 20, n. 16, p. 2812–2820, 2004. Disponível em: <<https://doi.org/10.1093/bioinformatics/bth335>>.
- DUVAL, J. Factorizing words over an ordered alphabet. **J. Algorithms**, v. 4, n. 4, p. 363–381, 1983. Disponível em: <[https://doi.org/10.1016/0196-6774\(83\)90017-2](https://doi.org/10.1016/0196-6774(83)90017-2)>.
- FRANEK, F. et al. Algorithms to compute the lyndon array. **CoRR**, abs/1605.08935, 2016. Disponível em: <<http://arxiv.org/abs/1605.08935>>.
- GHUMAN, S. S.; GIAQUINTA, E.; TARHIO, J. Alternative algorithms for Lyndon factorization. In: **Proceedings of the Prague Stringology Conference**. [s.n.], 2014. p. 169–178. Disponível em: <<http://www.stringology.org/event/2014/p16.html>>.
- _____. Lyndon factorization algorithms for small alphabets and run-length encoded strings. **Algorithms**, v. 12, n. 6, p. 124, 2019. Disponível em: <<https://doi.org/10.3390/a12060124>>.
- GLEN, A.; SIMPSON, J.; SMYTH, W. F. Counting lyndon factors. **Electron. J. Comb.**, v. 24, n. 3, p. 3, 2017. Disponível em: <<https://doi.org/10.37236/6915>>.
- I, T. et al. Efficient lyndon factorization of grammar compressed text. **CoRR**, abs/1304.7061, 2013. Disponível em: <<http://arxiv.org/abs/1304.7061>>.
- LOTHAIRE, M. (Ed.). **Combinatorics on Words**. Cambridge University Press, 1997. Disponível em: <<https://doi.org/10.1017/cbo9780511566097>>.
- MANBER, U.; MYERS, E. W. Suffix arrays: A new method for on-line string searches. **SIAM J. Comput.**, v. 22, n. 5, p. 935–948, 1993.
- MANTACI, S. et al. Suffix array and Lyndon factorization of a text. **J. Discrete Algorithms**, v. 28, p. 2–8, 2014. Disponível em: <<https://doi.org/10.1016/j.jda.2014.06.001>>.
- NEUBURGER, S.; SOKOL, D. Succinct 2d dictionary matching. **Algorithmica**, v. 65, n. 3, p. 662–684, 2013. Disponível em: <<https://doi.org/10.1007/s00453-012-9615-9>>.
- SMYTH, B. **Computing Patterns in Strings**. Pearson Addison-Wesley, 2003. (ACM Press Bks). ISBN 9780201398397. Disponível em: <<https://books.google.com.br/books?id=iKR0EewiCu4C>>.
- TSURUTA, K. et al. Grammar-compressed self-index with Lyndon words. **CoRR**, abs/2004.05309, 2020. Disponível em: <<https://arxiv.org/abs/2004.05309>>.