UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I
CENTRO DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE MATEMÁTICA
CURSO DE LICENCIATURA EM MATEMÁTICA

JEFFERSON ALVES DA SILVA

GRAFOS DE CONTEXTOS DE SISTEMAS SIMBÓLICOS DE MEMÓRIA FINITA

CAMPINA GRANDE
2022

JEFFERSON ALVES DA SILVA

GRAFOS DE CONTEXTOS DE SISTEMAS SIMBÓLICOS DE MEMÓRIA FINITA

Trabalho de Conclusão de Curso apresentado ao Departamento de Matemática do Centro de Ciências e Tecnologia da Universidade Estadual da Paraíba como requisito parcial à obtenção do título de Licenciado em Matemática.

Área de concentração: Combinatória

Orientador: Prof. Dr. Vilmar Vaz da Silva

CAMPINA GRANDE
2022

S586g Silva, Jefferson Alves da.
Grafos de contextos de sistemas simbólicos de memória finita [manuscrito] / Jefferson Alves da Silva. - 2022.
35 p.

Digitado.
Trabalho de Conclusão de Curso (Graduação em Matemática) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia , 2022.
"Orientação : Prof. Dr. Vilmar Vaz da Silva , Coordenação do Curso de Computação - CCT."

1. Sequências com restrições. 2. Algoritmo. 3. Grafo mínimo. I. Título

21. ed. CDD 519

Elaborada por Talita M. A. Tavares - CRB - CRB 15/971 BC/UEPB

JEFFERSON ALVES DA SILVA

GRAFOS DE CONTEXTOS DE SISTEMAS SIMBÓLICOS DE MEMÓRIA FINITA

Trabalho de Conclusão de Curso apresentado ao Departamento de Matemática do Centro de Ciências e Tecnologia da Universidade Estadual da Paraíba como requisito parcial à obtenção do título de Licenciado em Matemática.

Área de concentração: Combinatória

Aprovado em: 22/11/2022.

BANCA EXAMINADORA

Vilmar Vaz da Silva

Prof. Dr. Vilmar Vaz da Silva (Orientador)
Universidade Estadual da Paraíba (UEPB)

Maria Isabelle Silva Dias Yanes

Prof. Dra. Maria Isabelle Silva Dias Yanes
Universidade Estadual da Paraíba (UEPB)

Israel Buriti Galvão

Prof. Dr. Israel Buriti Galvão
Universidade Estadual da Paraíba (UEPB)

À minha querida família, DEDICO.

AGRADECIMENTOS

Quero agradecer a Deus pela grande proteção nessa trajetória, nesse momento tão importante, agradeço a minha familia e amigos por estarem ao meu lado durante quase 6 anos de curso, quero agradecer em particular aos meus amigos, Tonny Josias, Gilson Pacífico e Lucas Eduardo, que desde o ínicio, contribuiram diretamente com a minha formação especialmente, Tonny Josias que todas as noites durante um período viajava 40km me auxiliando com transporte.

Agradeço a minha mãe Severina Alves e Minha avó Maria Solidade que sempre me incentivaram e me deram um suporte tanto emocional quanto finceiro.

Agradeço também ao meu professor, orientador e amigo Dr.Vilmar Vaz da Silva, pela força e dedicação, que com muita sabedoria e paciência conseguimos concluir nosso trabalho.

Por fim, agradecer a todos os professores que me ajudaram a trilhar esse caminho e todo conhecimento que me foi trasmitido, assim, me tornando um professor.

RESUMO

O presente trabalho apresenta um breve estudo de sequências com restrições que têm sido usadas em muitos sistemas de armazenamento e codificação de dados. Essas sequências são descritas no trabalho por um grafo determinístico. O grafo com o menor número de vértices é obtido, em geral, por um procedimento de dois passos: o primeiro passo é gerar um grafo inicial e, o segundo, é aplicar um algoritmo de minimização de vértices para identificar classes de vértices equivalentes. Aplicando outro conceito do conjunto de restrições, este trabalho apresenta outro algoritmo de divisão para determinar os vértices da apresentação mínima sem determinar um grafo inicial. Assim, a função de transição que conecta os vértices é calculada sobre um conjunto menor, reduzindo os esforços computacionais para a construção deste grafo.

Palavras-chave: sequências com restrições; algoritmo; grafo mínimo.

ABSTRACT

This work presents a brief study of constrained sequences that have been used in many data storage and encoding systems. These sequences are described in the work by a deterministic graph. The graph with the fewest vertices is obtained, in general, by a two-step procedure: the first step is to generate an initial graph and the second is to apply a vertex minimization algorithm to identify equivalent classes of vertices. Applying another constraint set concept this work presents another division algorithm to determine the vertices of the minimal presentation without determining an initial graph. Thus, the transition function that connects the vertices is calculated on a smaller set, reducing the computational efforts for the construction of this graph.

Keywords: restricted sequences; algorithm; minimal graph.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

Página

1 INTRODUÇÃO

Um Sistema Simbólico Invariante por Deslocamento de Tipo Finito (SFT) é um sistema dinâmico simbólico especificado por um conjunto finito de palavras proibidas, denotado por $\mathcal{F}$, sobre um alfabeto finito $\mathcal{A}$, e todas as palavras sobre $\mathcal{A}$ sem fatores em $\mathcal{F}$ formam a linguagem L do SFT. O conjunto único minimal de palavras proibidas, denotado por $\mathcal{O}$, é chamado de conjunto de todas as primeiras proibições (Lind, 1995, p.33). Um SFT também pode ser especificado por um grafo direcionado rotulado, chamado de apresentação do SFT (Lind, 1995, Teorema 3.1.5). Sistemas práticos usados para armazenar ou transmitir informações codificadas de dados requerem a especificação de um conjunto $\mathcal{F}$ Immink (1995) e Marcus (1999).

O conjunto de contextos de uma palavra $\mathbf{w}$ é o conjunto de todas as palavras na linguagem L que podem seguir $\mathbf{w}$. Se duas palavras têm conjuntos seguidores iguas, elas são chamadas de palavras equivalentes pelo conjunto de contextos ou simplesmente, palavras equivalentes. Os possíveis conjuntos de contextos de um SFT permitem uma apresentação de forma geométrica, ou seja, por meio de um grafo de contextos, que é a apresentação determinística única com o menor número de vértices de um SFT irredutível, isto é, com a menor quantidade de estados. Essa apresentação mínima é obtida, em geral, por meio de um procedimento de duas etapas: a primeira etapa é gerar uma apresentação determinística inicial do conjunto $\mathcal{F}$ (alguns métodos são propostos em (Lind, 1995, Teorema 3.1.5), Crochemore (1998)). De acordo com Crochemore (1998) o conjunto de vértices da apresentação inicial é obtido da linguagem antifatorial associada à linguagem L de um SFT. Então uma função de transição conecta os vértices e rotula as arestas. Em alguns casos especiais, este método produz a apresentação mínima Manada (2006).

O segundo passo é aplicar um algoritmo de minimização de vértices Berstel (2010) para identificar vértices da apresentação inicial com o mesmo conjunto de contextos. O algoritmo de Moore, veja por exemplo (Lind, 1995, pág. 92), (Marcus, 1999, pág. 1661), tem complexidade assintótica n^2, em que n é o número de vértices da apresentação inicial. O algoritmo de Hopcroft com complexidade assintótica $n \log n$ é apresentado em Hopcroft (1971). Uma abordagem alternativa é proposta em Chaves (2006), que determina inicialmente o conjunto de vértices da apresentação mínima sem determinar um grafo inicial, e apenas então calcula a função de transição. Essa abordagem permite uma redução potencial na complexidade, uma vez que a função de transição é calculada apenas para um conjunto reduzido de vértices enquanto em métodos convencionais isso é calculado para todos os vértices da apresentação inicial. Neste sentido, este trabalho apresenta três resultados importantes. Em primeiro lugar, identifica-se um conjunto necessário e suficiente de palavras $\mathcal{W}$ que representam todos os conjuntos de contextos possíveis da linguagem de um SFT. Segundo, formaliza-se a ideia introduzida em Chaves (2006),

em que o conjunto de contextos de palavras equivalentes são identificados através dos subconjuntos mínimos de palavras em L que não podem segui-las, ou seja, conjuntos de restrições. Terceiro, é proposto um algoritmo de divisão para particionar $\mathcal{W}$ em classes de palavras com o mesmo conjunto de contextos que formam os vértices do grafo de contextos. Essa abordagem de divisão é realizada com um algoritmo recursivo do tipo Moore derivado de dois novos conceitos introduzidos em Chaves (2014), as chamadas máscara de restrição e memória de restrição. Alternativa ao método apresentado em Chaves (2006) que mescla palavras em $\mathcal{W}$ com o mesmo conjunto de contextos, logo a proposta apresentada neste trabalho é por divisão, que normalmente é menos complexa do que fusão Berstel (2010).

O trabalho está organizado em cinco capítulos. O Capítulo 2 contém material de fundo sobre dinâmica simbólica e partição. O Capítulo 3 formaliza a noção de conjuntos de restrições e suas propriedades e apresenta os conceitos de máscara de restrição e de memória de restrição que é o núcleo do mecanismo recursivo de particionamento proposto no Capítulo 4. O Capítulo 5 resume as conclusões deste trabalho.

2 PRELIMINARES

Neste capítulo serão explorados conceitos e resultados básicos sobre dinâmica simbólica, grafos e partições Lind (1995), Berstel (2010), necessários para o desenvolvimento do trabalho.

2.1 Dinâmica Simbólica

A teoria de dinâmica simbólica é usada como ferramenta matemática na teoria de códigos e teoria da informação. Nesta seção são apresentados alguns conceitos desta teoria, vinculados a sistemas dinâmicos simbólicos de memória finita.

Seja $\mathcal{A}^{\mathbb{Z}}$ o conjunto de sequências bi-infinitas $x = (x_i)_{i\in\mathbb{Z}} = \cdots x_{-2}x_{-1}x_0x_1\cdots$ de símbolos de um alfabeto finito $\mathcal{A}$. Uma sequência finita de símbolos consecutivos $\mathbf{w} = a_1a_2\cdots a_n,\ a_i \in \mathcal{A},\ 1 \leqslant i \leqslant n$ é chamada uma palavra ou bloco em $\mathcal{A}$ de comprimento n. Uma palavra $\mathbf{w}$ é um fator de um ponto $x \in \mathcal{A}^{\mathbb{Z}}$, denotado por $\mathbf{w} \prec x$, se existem inteiros $i \leq j$, tais que $\mathbf{w} = x_ix_{i+1}\cdots x_j$. Escreve-se $\mathbf{w} \prec_i x$ para enfatizar que $\mathbf{w}$ é um fator de x iniciando no índice i. A noção de fator de um ponto x pode ser naturalmente estendida para a noção de fator de uma palavra e usar-se-á a mesma notação. O conjunto de todas as palavras sobre $\mathcal{A}$ incluindo a palavra vazia ε que satisfaz $\mathbf{w}\varepsilon = \varepsilon\mathbf{w} = \mathbf{w}$ é $\mathcal{A}^*$, em que $\mathbf{w}\varepsilon$ é a concatenação da palavra $\mathbf{w}$ com a palavra vazia ε.

A aplicação deslocamento $\sigma : \mathcal{A}^{\mathbb{Z}} \to \mathcal{A}^{\mathbb{Z}}$ é definida por $\sigma(x) = y$ com $y_i = x_{i+1}$. Observe que σ é inversível e σ^{-1} é tal que $\sigma^{-1}(y) = x$ com $x_i = y_{i-1}$. Denota-se por σ^k $(k > 0)$ a composição de σ por ela mesma k-vezes, ou seja,

$$\sigma^k = \underbrace{\sigma \circ \cdots \circ \sigma}_{k-\text{vezes}}.$$

Seja $\mathcal{F}$ uma coleção de palavras sobre $\mathcal{A}$ e seja $\mathsf{X}_{\mathcal{F}}^{\mathcal{A}}$ o subconjunto de $\mathcal{A}^{\mathbb{Z}}$ formado por sequências bi-infinitas que não possuem como fator uma palavra de $\mathcal{F}$, neste contexto, $\mathcal{F}$ é referido como uma lista de palavras proibidas. Um Sistema Dinâmico Simbólico (SS) é um conjunto $X = \mathsf{X}_{\mathcal{F}}^{\mathcal{A}}$. Se no contexto não houver confusão sobre o alfabeto usado, utiliza-se a notação mais simples $X = \mathsf{X}_{\mathcal{F}}$. Quando $\mathcal{F}$ é finito, diz-se que X é de tipo finito e denota-se por SFT (*shift of finite type*). É importante observar que um SS X é invariante por deslocamento, ou seja, $\sigma(X) = X$.

Exemplo 2.1. Seja X o conjunto de todas as sequências binárias em que não ocorrem dois 1's consecutivos. Daí, $X = \mathsf{X}_{\mathcal{F}}$, em que $\mathcal{F} = \{11\}$. Observe que este espaço é um SFT.

Denota-se por $\mathcal{B}_n(X)$ o conjunto de todas as palavras de comprimento n que ocorrem em pontos de um SS X e a linguagem de X é a coleção

$$L = \mathcal{B}(X) = \bigcup_{n=0}^{\infty} \mathcal{B}_n(X),$$

em que $\mathcal{B}_0(X) = \varepsilon$, ou seja, L é o conjunto de todas as palavras que ocorrem em pontos de X, incluindo a palavra vazia ε. É importante observar que a linguagem de um SS X é fatorial e prolongável, isto é, se $\mathbf{w} \in L$ então todo fator de $\mathbf{w}$ também está em L, além disso, existem palavras não vazias $\mathbf{u}$ e $\mathbf{v}$ em L tal que $\mathbf{uwv} \in L$. O resultado a seguir afirma que um SS é completamente determinado pela sua linguagem, o que garante que dois SSs são iguais se, e somente se, possuem a mesma linguagem.

Proposição 2.1. *(1) Seja X um SS e $L = \mathcal{B}(X)$ sua linguagem. Se $\mathbf{w} \in L$, então:*

(a) todo sub-bloco de $\mathbf{w}$ pertence à L e

(b) existem blocos não vazios $\mathbf{u}, \mathbf{v} \in L$ tais que $\mathbf{uwv} \in L$.

(2) As linguagens de SS's são caracterizadas por (1). Isto é, se L é uma coleção de blocos sobre $\mathcal{A}$, então $L = \mathcal{B}(X)$ para algum SS X se, e somente se, L satisfaz a condição (1).

(3) A linguagem de um SS determina o mesmo. De fato, para algum SS, $X = X_{(\mathcal{B}(X))^C}$. Assim dois SS são iguais se, e somente se, possuem a mesma linguagem.

Demonstração. (1) Se $\mathbf{w} \in L = \mathcal{B}(X)$, então $\mathbf{w}$ ocorre em algum ponto $x \in X$. Mas então, todo sub-bloco de $\mathbf{w}$ também ocorre em x e daí está em L. Além disso, claramente existem blocos não-vazios $\mathbf{u}$ e $\mathbf{v}$ tais que $\mathbf{uwv}$ ocorre em x, pois, se assim não fosse, teríamos $x = \mathbf{w}$, o que é absurdo, pois, x é uma sequência bi-infinita. Daí, $\mathbf{u}, \mathbf{v} \in L$ e $\mathbf{uwv} \in L$.

(2) Sejam L uma coleção de blocos satisfazendo (1) e X denotar o SS X_{L^C}, vamos mostrar que $L = \mathcal{B}(X)$.

Seja $\mathbf{w} \in \mathcal{B}(X) = \mathcal{B}(X_{L^C})$. Portanto, $\mathbf{w}$ ocorre em algum ponto de X_{L^C}, o que garante que $\mathbf{w} \notin L^C$, ou seja, $\mathbf{w} \in L$. Daí, $\mathcal{B}(X) \subseteq L$. Por outro lado, suponha que

$$\mathbf{w} = x_0\, x_1 \cdots x_m \in L.$$

Então, por repetidas aplicações de (1b), podemos encontrar símbolos x_j com $j > m$ e x_i com $i < 0$ também que por (1a) todo sub-bloco de $x = (x_i)_{i\in\mathbb{Z}}$ está em L. Isto significa que $x \in X_{L^C}$. Desde que $\mathbf{w}$ ocorre em x, temos que

$$\mathbf{w} \in \mathcal{B}(X_{L^C}) = \mathcal{B}(X),$$

provando que $L \subseteq \mathcal{B}(X)$.

(3) Se $x \in X$, nenhum bloco ocorrendo em x está em $(\mathcal{B}(X))^C$. Daí,

$$x \in X_{(\mathcal{B}(X))^C} \Rightarrow X \subseteq X_{(\mathcal{B}(X))^C}.$$

Por outro lado, desde que X é um SS, existe uma coleção $\mathcal{F}$ tal que $X = X_{\mathcal{F}}$. Agora, se $x \in X_{(\mathcal{B}(X))^C}$, então todo bloco em x deve estar em $\mathcal{B}(X) = \mathcal{B}(X_{\mathcal{F}})$ e também não pode estar em $\mathcal{F}$. Portanto,

$$x \in X_{\mathcal{F}} \Rightarrow X_{(\mathcal{B}(X))^C} \subseteq X_{\mathcal{F}} = X.$$

□

Definição 2.1. Um SS X com linguagem L é irredutível se para todo par ordenado de palavras $\mathbf{u}, \mathbf{v} \in L$ existe $\mathbf{w} \in L$ tal que $\mathbf{uwv} \in L$.

Observe que o SFT do Exemplo 2.1 é irredutível, pois quaisquer duas palavras da linguagem conectadas por uma sequência de zeros gera uma palavra que pertence a linguagem.

Definição 2.2. O contexto à direita de uma palavra $\mathbf{w} \in L$, denotado por $F(\mathbf{w})$, é o conjunto de todas as palavras em L que quando concatenadas a $\mathbf{w}$ pela direita, forma-se uma palavra em L, isto é,

$$F(\mathbf{w}) = \{\mathbf{u} \in L : \mathbf{wu} \in L\}$$

Duas palavras $\mathbf{w}, \mathbf{u}$ com o mesmo contexto à direita $F(\mathbf{w}) = F(\mathbf{u})$ são ditas equivalentes. Uma palavra $\mathbf{w} = w_1 \cdots w_n$ é uma proibicão mínima se $\mathbf{w} \notin L$ e ambas $w_1 \cdots w_{n-1} \in L$ e $w_2 \cdots w_n \in L$. Denota-se por $\mathcal{O}$ o conjunto formado por todas as palavras proibidas mínimas de um SS. Devido à unicidade de $\mathcal{O}$ (Lind, 1995, p. 12), quando X é um SFT pode-se definir a memória de X.

Definição 2.3. Seja X um SFT com conjunto de palavras proibidas mínimas $\mathcal{O}$. A memória de X é definida por

$$m = \underset{\mathbf{u} \in \mathcal{O}}{\text{Máx}}\{|\mathbf{u}| - 1\}.$$

É importante observar que a minimalidade de $\mathcal{O}$ garante que a linguagem formada por este conjunto é antifatorial (Crochemore (1998)), ou seja, $\forall \mathbf{u}, \mathbf{v} \in \mathcal{O}$ com $\mathbf{u} \neq \mathbf{v}$, então, $\mathbf{u}$ não é um fator de $\mathbf{v}$.

Seja $\mathbf{u} \in \mathcal{A}^*$, o conjunto de prefixos de $\mathbf{u}$ é definido como $\mathcal{P}(\mathbf{u}) = \{\mathbf{v} | \exists\ \mathbf{t} \in \mathcal{A}^*$ com $\mathbf{vt} = \mathbf{u}\}$, como uma extensão natural, se $M \subseteq \mathcal{A}^*$ então $\mathcal{P}(M) = \underset{\mathbf{u} \in M}{\cup} \mathcal{P}(\mathbf{u})$. Similarmente, o conjunto de sufixos de $\mathbf{u}$ é definido como $\mathcal{S}(\mathbf{u}) = \{\mathbf{v} | \exists\ \mathbf{t} \in \mathcal{A}^*$ com $\mathbf{tv} = \mathbf{u}\}$ e para um subconjunto arbitrário M de $\mathcal{A}^*$, $\mathcal{S}(M) = \underset{\mathbf{u} \in M}{\cup} \mathcal{S}(\mathbf{u})$. Observe que $\mathbf{u} \in \mathcal{P}(\mathbf{u}), \mathcal{S}(\mathbf{u})$, pois $\varepsilon \in \mathcal{A}^*$ e $\mathbf{u}\varepsilon = \varepsilon\mathbf{u} = \mathbf{u}$; daí, os conjuntos dos prefixos e sufixos próprios de $\mathbf{u}$ são dados, respectivamente, por $\mathcal{P}(\mathbf{u}) \setminus \{\mathbf{u}\}$ e $\mathcal{S}(\mathbf{u}) \setminus \{\mathbf{u}\}$. A extensão de uma palavra $\mathbf{w} \in \mathcal{A}^*$ por outra palavra $\mathbf{v} \in \mathcal{A}^*$ é $\mathbf{wv}$.

Exemplo 2.2. seja $\mathbf{w} = abbc$, então

$$\mathcal{P}(\mathbf{w}) = \{\varepsilon,\ a,\ ab,\ abb,\ abbc\} \quad \text{e} \quad \mathcal{S}(\mathbf{w}) = \{\varepsilon,\ c,\ bc,\ bbc,\ abbc\}$$

.

Dados dois conjuntos $B_1, B_2 \subseteq \mathcal{A}^*$, define-se:

$$\begin{cases} B_1 B_2^{-1} = \{\mathbf{w}_1 \in \mathcal{A}^* | \ \mathbf{w}_1 \mathbf{w} \in B_1 \text{ para } \mathbf{w} \in B_2\} \\ B_1^{-1} B_2 = \{\mathbf{w}_2 \in \mathcal{A}^* | \ \mathbf{w} \mathbf{w}_2 \in B_2 \text{ para } \mathbf{w} \in B_1\} \end{cases}.$$

Exemplo 2.3. Para $B_1 = \{adbbc, bdaa\}$ e $B_2 = \{aa, adbbcdd\}$, tem-se

$$\begin{cases} \mathcal{P}(B_1) = \{\varepsilon, a, b, ad, bd, adb, bda, adbb, bdaa, adbbc\} \\ \mathcal{S}(B_2) = \{\varepsilon, a, d, aa, dd, cdd, bcdd, bbcdd, dbbcdd, adbbcdd\} \\ B_1 B_2^{-1} = \{bd\} \\ B_1^{-1} B_2 = \{dd\} \end{cases}.$$

2.1.1 Grafos

Definição 2.4. Dados $\mathcal{V}$ um conjunto de vértices, $\mathcal{E}$ um conjunto de arestas (ligações entre vértices) e $\mathcal{L} : \mathcal{E} \to \mathcal{A}$ uma função de rotulação, então

$$G = (\mathcal{V}, \mathcal{E}, \mathcal{L})$$

é um grafo direcionado rotulado (doravante denominado de grafo). As funções $i : \mathcal{E} \to \mathcal{V}$ e $t : \mathcal{E} \to \mathcal{V}$ especificam o vértice inicial e o vértice final de uma aresta, respectivamente.

Definição 2.5. Uma trajetória em G é uma sequência de arestas $\pi = e_1 e_2 \cdots e_n$, tal que, o vértice terminal de e_i é o vértice inicial de e_{i+1}. Essa trajetória será denominada de circuito se todas as arestas e_i, $i = 1, 2, \ldots, n$ são distintas e o vértice terminal de e_n é o vértice inicial de e_1. O rótulo de π é a palavra $\mathcal{L}(\pi) = \mathcal{L}(e_1)\mathcal{L}(e_2)\cdots\mathcal{L}(e_n)$. Um caminho bi-infinito em G é uma sequência bi-infinita de arestas $\xi = \cdots e_{-1} e_0 e_1 \cdots$ tal que $t(e_i) = i(e_{i+1})$ para todo i.

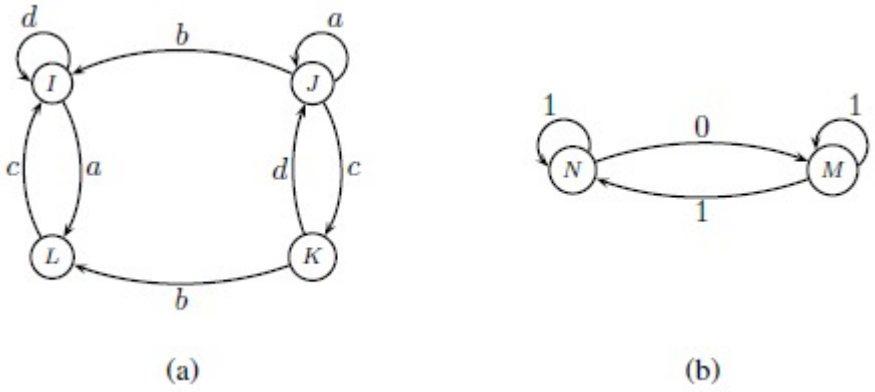
Observação 2.1. Uma palavra $\mathbf{w} \in L$ é gerada por uma trajetória π em G se $\mathbf{w} = \mathcal{L}(\pi)$.

Definição 2.6. Uma aresta é um laço se seus extremos coincidem e uma ligação caso contrário.

Definição 2.7. O grau de um vértice J é o número de arestas que incidem em J , em que os laços são contados duas vezes. O grau de J em um grafo G será denotado por $g_G(J)$, ou simplesmente por $g(J)$.

A Figura 2.1.1 mostra dois grafos, em que o grafo em (a) possui 4 vértices, 8 arestas, a aresta JJ com rótulo a é um laço e $g(J) = 4$, e o grafo em (b) possui 2 vértices, 4 arestas, a aresta NN com rótulo 1 é um laço e $g(N) = 3$. Observe que *daacbcddda* e 111010 são palavras geradas por trajetórias nos grafos em 2.1.1(a) e 2.1.1(b), respectivamente.

Figura 2.1.1 – *Exemplo de Grafos.*



Fonte: Lind (1995).

Um vértice $I \in \mathcal{V}$ é dito não-essencial se nenhuma das arestas de G inicia ou termina em I. Um grafo é essencial se não possui vértices não-essenciais.

Definição 2.8. Um grafo G é chamado determinístico se para todos $e_1, e_2 \in \mathcal{E}$, $i(e_1) = i(e_2)$ e $\mathcal{L}(e_1) = \mathcal{L}(e_2)$, então $e_1 = e_2$.

Definição 2.9. Um grafo G é dito irredutível (ou conexo) se para todo par ordenado de vértices I e J existe uma trajetória em G iniciando em I e terminando em J.

Para os grafos da Figura 2.1.1, tem-se que ambos são essenciais; (a) é determinístico e redutível (não existe trajetória iniciando em L e terminando em J) , enquanto que (b) não é determinístico (arestas distintas iniciando em M com o mesmo rótulo 1), mas é irredutível.

Sejam G e G' grafos. Um homomorfismo de grafos de G em G' consiste em um par de aplicações $\Phi : \mathcal{V}(G) \to \mathcal{V}(G')$ e $\Psi : \mathcal{E}(G) \to \mathcal{E}(G')$ tais que

$$i(\Psi(e)) = \Phi(i(e)) \quad \text{e} \quad t(\Psi(e)) = \Phi(t(e)), \forall\ e \in \mathcal{E}(G).$$

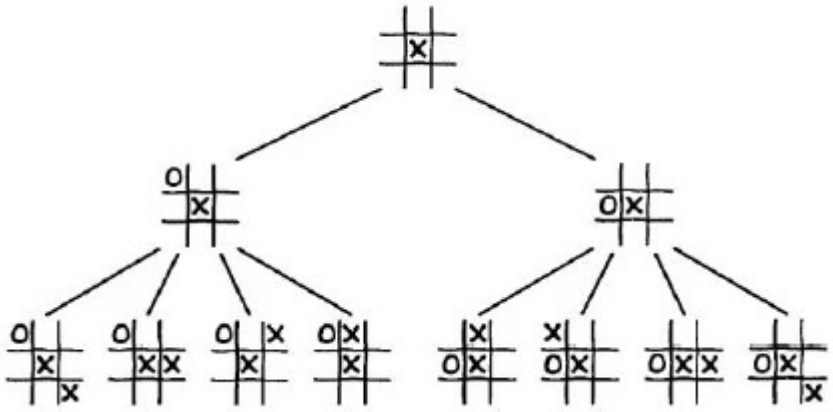
Neste caso, escreve-se $(\Phi, \Psi) : G \to G'$. Este homomorfismo é um isomorfismo de grafos se ambas Φ e Ψ são bijetivas. Dois grafos G e G' são isomorfos, denotados por $G \cong G'$, se existe um isomorfismo de grafos entre eles.

2.1.2 Árvores

Definição 2.10. Uma "árvore"é um grafo conexo que não possui circuitos.

Exemplo 2.4 (Árvore derivada de um jogo)**.** Suponha que estando em uma fase de um jogo deseja-se passar para a próxima fase, a partir daí é possível construir uma árvore da

Figura 2.1.2 – *Trecho da Árvore do Jogo da Velha.*



Fonte: Oliveira (2013).

seguinte maneira:

$$\begin{cases} \textbf{Vértices}: & \text{os estados do jogo.} \\ \textbf{Arestas}: & \text{existe uma aresta entre um estado do jogo e um estado que pode ser obtido através deste.} \end{cases}$$

A Figura 2.1.2 mostra um trecho da árvore do jogo da velha.

Teorema 2.1. *Um grafo G é uma árvore se, e somente se, existir um e apenas um caminho entre cada par de vértices.*

Demonstração. [$\Rightarrow$] Se G é uma árvore, então, por definição, G é conexo e sem circuitos. Como G é conexo, então existe um caminho entre cada par de vértices. Precisa-se mostrar que este caminho é único. Suponha que existam dois caminhos distintos entre um par de vértices. Ora, se existem dois caminhos distintos entre um par de vértices então a união destes caminhos contém um circuito. Mas por hipótese, o grafo não possui circuitos, portanto, existe apenas um caminho entre cada par de vértices.
[$\Leftarrow$] Mostrar-se-á que se existe um, e apenas um, caminho entre cada par de vértices, então G é uma árvore. Como existe um caminho entre cada par de vértices, então que G é conexo. Suponha que G contenha um circuito. A existência de um circuito no grafo implica que existe pelo menos um par de vértices a, b tais que existem dois caminhos distintos entre a e b. Porém, por hipótese, existe um, e apenas um caminho entre cada par de vértices e, portanto, o grafo não tem circuitos. Por definição um grafo conexo e sem circuitos é uma árvore. □

Teorema 2.2. *Seja $G(\mathcal{V}, \mathcal{E})$ um grafo com n vértices. As seguintes afirmações são equivalentes:*

a) G é uma árvore;

b) G é conexo e possui $n-1$ arestas;

c) G possui $n-1$ arestas e não possui circuitos;

d) Existe exatamente um caminho entre cada par de vértices;

e) G não contém circuitos, e para todo $v,\ w \in \mathcal{V}$, a adição da aresta $(v,\ w)$ produz no grafo exatamente um circuito.

Observação 2.2. Qualquer uma das afirmações do Teorema 2.2 podem ser usadas como definição de uma árvore.

2.1.3 Sistemas Simbólicos Regulares

Dado um grafo cujos ramos são rotulados com símbolos de um alfabeto $\mathcal{A}$ pode-se formar um subconjunto de $\mathcal{A}^{\mathbb{Z}}$, em que seus pontos são formados pela leitura dos rótulos dos caminhos bi-infinitos sobre o grafo. Estes conjuntos formam sistemas que modelam sequências usadas em armazenamento e transmissão de informações (Lind (1995); Immink (1995)).

Um SS X é representado por G, ou G é uma apresentação de X, se o rótulo de todo caminho bi-infinito em G é um elemento de X, com o contrário também verdadeiro. Um subconjunto $X = \mathsf{X}_G$ de $\mathcal{A}^{\mathbb{Z}}$ é um Sistema Simbólico Regular (SSR) se é representado por algum grafo G. SSRs são SSs (Lind, 1995, Teorema 3.1.4).

Teorema 2.3. *(Lind, 1995, Teorema 3.2.10) Um SS é um SSR se, e somente se, ele tem um número finito de contextos à direita.*

2.2 Partições

Uma coleção $\mathcal{P}$ de subconjuntos disjuntos não-vazios de um conjunto $\mathcal{D}$ é chamada de uma partição se satisfaz $\mathcal{D} = \bigcup_{P \in \mathcal{P}} P$. Os conceitos de partição e relação de equivalência são interligados Dummit (2004). Seja $\mathcal{Q}$ outra partição de $\mathcal{D}$. Então, $\mathcal{Q}$ é chamada um refinamento de $\mathcal{P}$ ou $\mathcal{P}$ é a partição mais densa de $\mathcal{Q}$, se cada classe de $\mathcal{Q}$ está contida em alguma classe de $\mathcal{P}$. Dadas as partições $\mathcal{P}$ e $\mathcal{Q}$, denota-se por $\mathcal{P} \wedge \mathcal{Q}$ a partição mais densa que refina ambas $\mathcal{P}$ e $\mathcal{Q}$ e os elementos correspondentes são os conjuntos não-vazios $P \cap Q$ para todo $P \in \mathcal{P}$ e $Q \in \mathcal{Q}$. Esta notação é estendida para algum número finito de partições de $\mathcal{D}$ tal que $\mathcal{P} = \mathcal{P}_1 \wedge \mathcal{P}_2 \wedge \cdots \wedge \mathcal{P}_n$ é o refinamento mais denso das partições $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n$.

Exemplo 2.5. Seja $\mathcal{D} = \{\varepsilon, b, c, bc, abc, cbda\}$ e $\mathcal{P} = \{\{\varepsilon, abc\}, \{b, c, bc, cbda\}\}$, $\mathcal{Q} = \{\{\varepsilon, b, cbda\}, \{c, bc, abc\}\}$ duas partições, então:

$$\begin{cases} \{\varepsilon, abc\} \cap \{\varepsilon, b, cbda\} = \{\varepsilon\} \\ \{\varepsilon, abc\} \cap \{c, bc, abc\} = \{abc\} \end{cases} \quad \text{e} \quad \begin{cases} \{b, c, bc, cbda\} \cap \{\varepsilon, b, cbda\} = \{b, cbda\} \\ \{b, c, bc, cbda\} \cap \{c, bc, abc\} = \{c, bc\} \end{cases}.$$

Daí, $\mathcal{P} \wedge \mathcal{Q} = \{\{\varepsilon\}, \{abc\}, \{b, cbda\}, \{c, bc\}\}$.

3 CONJUNTO DE RESTRIÇÕES

Neste capítulo serão estudados os conceitos de conjunto de restrições, máscara de restrição e memória de restrição apresentados em Chaves (2014), que servirão de base para o algorítmo apresentado no próximo capítulo para a construção do grafo de contextos de um SFT.

Definição 3.1. Seja $\mathbf{w} \in L$. O conjunto de restrições de $\mathbf{w}$, denotado por $\mathcal{C}(\mathbf{w})$ é dado por

$$\mathcal{C}(\mathbf{w}) \triangleq \{\mathbf{v} \in L : \ \mathbf{wv} \notin L, \text{ mas, } \mathbf{wu} \in L \ \forall \ \mathbf{u} \in \mathcal{P}(\mathbf{v}) \setminus \{\mathbf{v}\}\}.$$

Um fato importante a ser considerado é que o contexto à direita de uma palavra na linguagem é unicamente caracterizado pelo seu conjunto de restrições, como mostra o resultado a seguir.

Teorema 3.1. *Quaisquer que sejam* $\mathbf{w}, \ \mathbf{w}' \in L$, $F(\mathbf{w}) = F(\mathbf{w}')$ *se, e somente se,* $\mathcal{C}(\mathbf{w}) = \mathcal{C}(\mathbf{w}')$.

Demonstração. Suponha que $F(\mathbf{w}) = F(\mathbf{w}')$. Daí $\mathbf{v} \notin F(\mathbf{w})$ se, e somente se, $\mathbf{v} \notin F(\mathbf{w}')$, para $\mathbf{v} \in L$. Como L é fatorial e prolongável, $\mathbf{v}$ deve ter o prefixo mais curto $\mathbf{u} \in L$ satisfazendo $\mathbf{u} \notin F(\mathbf{w}')$, o que implica que $\mathcal{P}(\mathbf{u})\backslash\{\mathbf{u}\} \subseteq F(\mathbf{w})$ e então, $\mathbf{u} \in \mathcal{C}(\mathbf{w})$. Consequentemente, $\mathbf{u} \in F(\mathbf{w}')$ e $\mathcal{P}(\mathbf{u})\backslash\{\mathbf{u}\} \subseteq F(\mathbf{w}')$, e então $\mathbf{u} \in \mathcal{C}(\mathbf{w}')$. Concluimos que $\mathcal{C}(\mathbf{w}) \subseteq \mathcal{C}(\mathbf{w}')$. Da mesma forma, pode-se mostrar que $\mathcal{C}(\mathbf{w}') \subseteq \mathcal{C}(\mathbf{w})$, e assim $\mathcal{C}(\mathbf{w}) = \mathcal{C}(\mathbf{w}')$.

Agora, suponha que $\mathcal{C}(\mathbf{w}) = \mathcal{C}(\mathbf{w}')$ e seja $\mathbf{z} \in F(\mathbf{w})$. Então $\mathbf{wz} \in L$ e da propriedade fatorial da linguagem, $\mathbf{u} \notin \mathcal{C}(\mathbf{w})$ para todo $\mathbf{u} \in \mathcal{P}(\mathbf{z})$. E se $\mathbf{z} \notin F(\mathbf{w}')$, existe $\mathbf{z}'$ o menor fator de $\mathbf{z}$, satisfazendo tanto $\mathbf{z}' \notin F(\mathbf{w}')$ e $\mathcal{P}(\mathbf{z}')\backslash\{\mathbf{z}'\} \in F(\mathbf{w}')$, o que implica que $\mathbf{z}' \in \mathcal{C}(\mathbf{w}')$ e $\mathbf{z}' \in \mathcal{C}(\mathbf{w})$. Desta forma $\mathbf{z}' \notin F(\mathbf{w})$, desde $\mathbf{wz}'$ é um fator de $\mathbf{wz}$, uma contradição. De forma similar, e se $\mathbf{z} \in F(\mathbf{w}')$, logo, $\mathbf{z} \in F(\mathbf{w})$. Portanto, se $\mathcal{C}(\mathbf{w}) = \mathcal{C}(\mathbf{w}')$ então $F(\mathbf{w}) = F(\mathbf{w}')$. Concluimos que $\mathcal{C}(\mathbf{w}) = \mathcal{C}(\mathbf{w}')$ se, e somente se, $F(\mathbf{w}) = F(\mathbf{w}')$. □

O próximo teorema estabelece que o conjunto $\mathcal{P}(\mathcal{OA}^{-1})$ constitui a informação necessária e suficiente para identificar o conjunto de restrições de qualquer palavra na linguagem.

Teorema 3.2. *Para todo* $\mathbf{w} \in L$, $\mathcal{C}(\mathbf{w}) = \mathcal{C}(\mathbf{v})$ *para* $\mathbf{v}$ *o sufixo mais longo de* $\mathbf{w}$ *em* $\mathcal{P}(\mathcal{OA}^{-1})$.

Demonstração. Seja $\mathbf{v}$ o sufixo mais longo de $\mathbf{w}$ em $\mathcal{P}(\mathcal{OA}^{-1})$. Como $\mathbf{v} \in \mathcal{S}(\mathbf{w})$ e L é uma linguagem fatorial, para todo $\mathbf{u} \in L$ tal que $\mathbf{vu} \notin L$ segue que $\mathbf{wu} \notin L$ e, portanto, $F(\mathbf{w}) \subseteq F(\mathbf{v})$.

Agora, suponha que $F(\mathbf{v}) \nsubseteq F(\mathbf{w})$. Então existe $\mathbf{u} \in L$ tal que $\mathbf{wu} \notin L$, mas $\mathbf{vu} \in L$. Seja $\mathbf{u}' \in \mathcal{P}(\mathbf{u})$ o prefixo mais curto tal que $\mathbf{wu}' \notin L$, e seja $\mathbf{w}' \in \mathcal{S}(\mathbf{w})$ a palavra mais curta

tal que $\mathbf{w'u'} \notin L$. Note que $\mathbf{w'u'} \in \mathcal{O}$, caso contrário, existe um fator próprio $\mathbf{s} \in \mathcal{O}$ de $\mathbf{w'u'}$ que não é fator de $\mathbf{w'}$ e nem fator de $\mathbf{u'}$, assim $\mathbf{w'} \in \mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$. Como $\mathbf{v}$ é o sufixo mais longo de $\mathbf{w}$ em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$, então $|\mathbf{v}| \geqslant |\mathbf{w'}|$ e também $\mathbf{w'}$ é um sufixo de $\mathbf{v}$. Da propriedade fatorial da linguagem $\mathbf{vu'} \notin L$, e assim $\mathbf{vu} \notin L$, uma contradição. Segue que $F(\mathbf{v}) \subseteq F(\mathbf{w})$, e também $F(\mathbf{v}) = F(\mathbf{w})$. Do Teorema 1, $\mathcal{C}(\mathbf{w}) = \mathcal{C}(\mathbf{v})$. □

Dos Teoremas 3.1 e 3.2 é possível concluir que duas palavras em L têm o mesmo conjunto de contextos se seus sufixos mais longos em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$ tiverem o mesmo conjunto de restrições. A proposição a seguir permite determinar o conjunto de restrições da extensão de uma palavra a partir da extensão do seu sufixo mais longo em $\mathcal{P}(\mathcal{O}A^{-1})$ pelo mesmo símbolo, resultado importante para a construção do grafo de contextos baseado no conceito de conjunto de restrição.

Proposição 3.1. *Seja* $\mathbf{w} \in L$. *Se* $\mathbf{v}$ *é o sufixo mais longo de* $\mathbf{w}$ *em* $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$ *então* $\mathcal{C}(\mathbf{va}) = \mathcal{C}(\mathbf{wa})$ *para algum* $a \in \mathcal{A}$.

Demonstração. Sejam $\mathbf{v}$ e $\mathbf{u}$ os sufixos mais longos em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$ de $\mathbf{w}$ e $\mathbf{w}a$, respectivamente. Se $|\mathbf{v}a| \geqslant |\mathbf{u}|$, então o resultado segue pelo Teorema 3.2, desde que $\mathbf{u}$ também é um sufixo de $\mathbf{v}a$. Agora, suponha que $|\mathbf{u}| > |\mathbf{v}a|$. Como ambos $\mathbf{u}$ e $\mathbf{v}a$ são sufixos de $\mathbf{w}a$, e $\mathbf{u} \in \mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$, daí $\mathbf{v}$ é um sufixo de $\mathbf{u}\mathcal{A}^{-1}$ e $|\mathbf{v}| < |\mathbf{u}\mathcal{A}^{-1}|$. Como o prefixo de uma palavra em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$ também pertence a $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$, então $\mathbf{u}\mathcal{A}^{-1}$ é um sufixo de $\mathbf{w}$ em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$ mais longo que $\mathbf{v}$, o que é uma contradição. Assim, os sufixos mais longos de $\mathbf{v}a$ e $\mathbf{w}a$ em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$ são iguais, concluindo a prova pelo Teorema 3.2. □

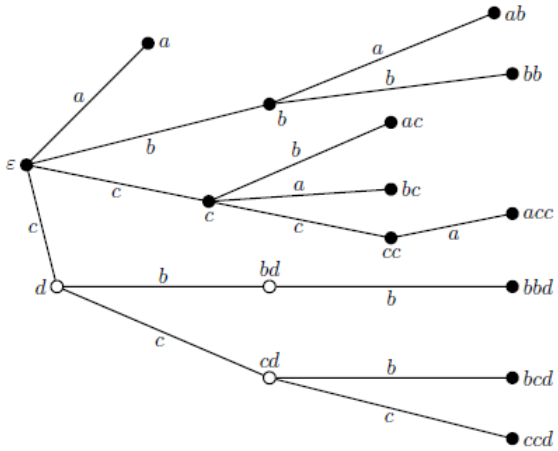
3.1 Máscara de Restrição

Existem essencialmente duas estratégias para particionar $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$: por fusão ou por divisão. Um algoritmo de mesclagem listaria e compararia o conjunto de restrições de cada palavra em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$. Normalmente, essa estratégia é computacionalmente mais complexa do que a divisão Berstel (2010). Esta seção introduz o conceito de máscara de restrição que juntamente com o conceito de memória de restrição que será visto na seção seguinte permitindo aplicar algoritmos de divisão para identificar palavras em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$ com conjuntos de contextos distintos.

Esses conceitos envolvem o cálculo do sufixo próprio mais longo em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$ dos elementos deste mesmo conjunto. Isso é tratado aqui como uma versão mais simples de um algoritmo de processamento de palavras sobre uma árvore que tem no máximo complexidade linear com a cardinalidade de $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$. Para um algoritmo relacionado veja Crochemore (1998).

A árvore deve ter o número mínimo de nós tal que o rótulo de cada nó é um sufixo de uma palavra em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$, e o rótulo do caminho da raiz ao nó seja o rótulo do nó lido da direita para a esquerda. Sempre que o rótulo de um nó é uma palavra em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$, este nó é marcado com um "•", caso contrário é marcado com um "∘". Para determinar

Figura 3.1.1 – *Árvore para determinar os sufixos próprios mais longos em* $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$.



Fonte: Elaborado pelo autor, 2022.

o sufixo próprio mais longo de uma palavra $\mathbf{w}$ em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$, basta seguir o caminho com o nó terminal rotulado $\mathbf{w}$ e registrar o último nó com a marca "•"antes desse caminho. O rótulo deste nó intermediário é o sufixo próprio mais longo de $\mathbf{w}$.

Exemplo 3.1. Seja

$$\mathcal{O} = \{abc,\ acd,\ accb,\ ccdb,\ bcdd,\ bbdd\}.$$

Temos que:

$$\mathcal{P}(\mathcal{O}\mathcal{A}^{-1}) = \{\varepsilon,\ a,\ b,\ c,\ ,ab,\ ac,\ bb,\ bc,\ cc,\ acc,\ bcd,\ bbd,\ ccd\}.$$

A Figura 3.1.1 apresenta a árvore para $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$. Observe que os nós bc, acc e ccd são folhas dessa árvore, também são palavras em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$. O sufixo próprio mais longo de ccd na árvore, o nó cd, é marcado com "∘"e, portanto, não pertence a $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$. No entanto, o nó ccd é marcado com "•", sendo ccd uma palavra em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$.

Definição 3.2. $\mathbf{u} \in L$ é uma restrição de $\mathbf{w} \in L$ se $\mathbf{wu} \notin L$, ou equivalentemente se $\mathbf{u} \notin F(\mathbf{w})$.

Definição 3.3 (Máscara de Restrição)**.** Seja $\mathbf{w} \in \mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$. Os elementos da máscara de restrição de $\mathbf{w}$, $\mathcal{M}(\mathbf{w}) \subset \mathcal{A} \times \{\square,\ \blacksquare\}$, satisfazem as seguintes propriedades:

- $(a,\ \blacksquare) \in \mathcal{M}(\mathbf{w}) \ \Leftrightarrow\ \mathbf{w}a \notin L,\ a \in \mathcal{A}$;
- $(a,\ \square) \in \mathcal{M}(\mathbf{w}) \ \Leftrightarrow\ \mathbf{w}a \in L,$ para $a \in \mathcal{A}$ um prefixo próprio de $\mathbf{u} \in L$ satisfazendo $\mathbf{wu} \notin L$;

- Se um símbolo não satisfaz as declarações anteriores, então não existe um par ordenado em $\mathcal{A} \times \{\square,\ \blacksquare\}$ com este símbolo.

Observação 3.1. A máscara de restrição identifica através das bandeiras $\square$ e $\blacksquare$, a informação que um símbolo $a \in \mathcal{A}$ transmite sobre as restrições de uma palavra $\mathbf{w} \in L$, mais especificamente, a bandeira $\square$ significa que a é um prefixo próprio de uma restrição de $\mathbf{w}$, e a bandeira $\blacksquare$ significa que a é ele mesmo uma restrição de $\mathbf{w}$.

Uma boa estratégia para calcular a máscara de restrição é indo das palavras mais curtas para as mais longas em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$. Para tanto, é importante seguir os seguintes passos:

1. seja $\mathbf{w} \in \mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$ e seja $\mathbf{v}$ o seu sufixo próprio mais longo em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$, pode-se começar listando $(a,\ \blacksquare) \in \mathcal{M}(\mathbf{w})$ para cada $a \in \mathbf{w}^{-1}\mathcal{O}$, e $(a,\ \square) \in \mathcal{M}(\mathbf{w})$ quando o símbolo a é um prefixo próprio de uma palavra em $\mathbf{w}^{-1}\mathcal{O}$;

2. incorporar em $\mathcal{M}(\mathbf{w})$ pares ordenados pertencentes à $\mathcal{M}(\mathbf{v})$ com os símbolos $a \in \mathcal{A}$ que ainda não apareceram em pares de $\mathcal{M}(\mathbf{w})$. Como $\mathbf{v}$ é mais curto que $\mathbf{w}$, o conjunto $\mathcal{M}(\mathbf{v})$ já foi calculado. Para a correção desse processo observe que se $(a,\ \square) \in \mathcal{M}(\mathbf{w})$ então $(a,\ \blacksquare) \notin \mathcal{M}(\mathbf{v})$, caso contrário deveria existir um elemento em $\mathcal{O}$ que também possui um fator próprio em $\mathcal{O}$, o que seria absurdo, pois, $\mathcal{O}$ é anti-fatorial. Também não se omite uma restrição de $\mathbf{w}$ em $\mathcal{M}(\mathbf{w})$ herdada de $\mathbf{v}$ por ignorar símbolos em $\mathcal{M}(\mathbf{v})$ que já apareciam em $\mathbf{w}^{-1}\mathcal{O}$;

3. Finalmente, para cada restrição $\mathbf{u}$ de $\mathbf{w}$ existe um sufixo $\mathbf{u}'$ de $\mathbf{w}$ em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$ e um prefixo $\mathbf{u}$" de $\mathbf{u}$ em $\mathcal{S}(\mathcal{A}^{-1}\mathcal{O})$ tal que $\mathbf{u}'\mathbf{u}$" pertence a $\mathcal{O}$. Como $\mathbf{v}$ é o mais longo de todos os sufixos próprios, $\mathcal{M}(\mathbf{v})$ contém todas as informações adicionais que não estão em $\mathbf{w}^{-1}\mathcal{O}$ para completar $\mathcal{M}(\mathbf{w})$.

Exemplo 3.2. Para $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$ dado no Exemplo 3.1, considere $\mathbf{w} = a$ em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$. Observe que

$$a^{-1}\mathcal{O} = \{bc,\ cd,\ ccb\},$$

portanto $\{(b,\ \square),\ (c,\ \square)\} \subseteq \mathcal{M}(a)$. Como o sufixo próprio mais longo de a em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$ é ε, e $\mathcal{M}(\varepsilon) = \varnothing$, concluímos que

$$\mathcal{M}(a) = \{(b,\ \square),\ (c,\ \square)\}.$$

Agora, considere $\mathbf{w} = acc$, que tem cc como seu sufixo mais longo em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$. Observe que

$$(acc)^{-1}\mathcal{O} = \{b\} \Rightarrow \{(b,\ \blacksquare)\} \subseteq \mathcal{M}(acc).$$

Pelo que já foi observado, a determinação das máscaras de c e de cc é um processo bem simples. Assim,

$$\mathcal{M}(c) = \{(c,\ \square)\} \quad \text{e} \quad \mathcal{M}(cc) = \{(c,\ \square),\ (d,\ \square)\}.$$

Por fim, devemos incorporar a $\mathcal{M}(acc)$ os pares $(c,\ \square)$ e $(d,\ \square)$. Logo,

$$\mathcal{M}(acc) = \{(b,\ \blacksquare),\ (c,\ \square),\ (d,\ \square)\}$$

.

Como uma extensão da definição de máscara de restrição, se $B \subseteq \mathcal{A}^*$ é tal que,

$$\forall\ \mathbf{u},\ \mathbf{w}\ \in\ B,\ \mathcal{M}(\mathbf{u})\ =\ \mathcal{M}(\mathbf{w}),$$

então define-se $\mathcal{M}(B)\ =\ \mathcal{M}(\mathbf{w})$.

3.2 Memória de Restrição

Nesta seção será estudado o conceito de memória de restrição. A importância primordial da memória de restrição é determinar as restrições de uma palavra $\mathbf{w} \in \mathcal{P}(\mathcal{OA}^{-1})$ estendida por um símbolo $a \in \mathcal{A}$, que é aplicado no algoritmo de divisão para particionar o conjunto $\mathcal{P}(\mathcal{OA}^{-1})$ bem como para determinar as transições de estados do grafo de contextos, como será demonstrado no Capítulo 4.

Definição 3.4 (Memória de Restrição)**.** Dada uma palavra $\mathbf{w} \in L$, a memória de restrição de $\mathbf{w}$, denotada por $\mathcal{R}(\mathbf{w})$, é o sufixo mais longo de $\mathbf{w}$ em $\mathcal{P}(\mathcal{OA}^{-1})$.

Observação 3.2. Segue do Teorema 3.2 que para todo $\mathbf{w} \in L$, se $\mathbf{v} = \mathcal{R}(\mathbf{w})$ então $\mathcal{C}(\mathbf{w}) = \mathcal{C}(\mathbf{v})$.

A partir da definição de máscara de restrição, quatro casos devem ser considerados para o cálculo da memória de restrição da extensão de uma palavra em $\mathcal{P}(\mathcal{OA}^{-1})$:

1) Para $(a,\ \blacksquare) \in \mathcal{M}(\mathbf{w})$ então $\mathcal{R}(\mathbf{w}a)$ não está definido, desde que $\mathbf{w}a \notin L$;

2) Se $\mathbf{w} = \varepsilon$, então $\mathcal{R}(\varepsilon a) = \mathcal{R}(a)$;

3) Para $(a,\ \square) \in \mathcal{M}(\mathbf{w})$ e $\mathbf{w}a \in \mathcal{P}(\mathcal{OA}^{-1})$, então $\mathcal{R}(\mathbf{w}a) = \mathbf{w}a$;

4) Para $(a,\ \square) \in \mathcal{M}(\mathbf{w})$ e $\mathbf{w}a \notin \mathcal{P}(\mathcal{OA}^{-1})$, a memória de restrição $\mathcal{R}(\mathbf{w}a)$ é o sufixo próprio mais longo de $\mathbf{w}a$ em $\mathcal{P}(\mathcal{OA}^{-1})$, o qual da Proposição 3.1 é a memória de restrição do sufixo próprio mais longo de $\mathbf{w}$ em $\mathcal{P}(\mathcal{OA}^{-1})$ estendido por a.

Neste último caso, a memória de restrição já foi calculada uma vez que ela é determinada das palavras mais curtas para as mais longas em $\mathcal{P}(\mathcal{OA}^{-1})$. Assim, a busca do sufixo próprio mais longo em $\mathcal{P}(\mathcal{OA}^{-1})$ limita a complexidade computacional para determinar

a memória de restrição. Essa tarefa pode ser realizada por algoritmos sobre uma árvore com no máximo complexidade linear com a cardinalidade de $\mathcal{P}(\mathcal{OA}^{-1})$.

Exemplo 3.3. Considere a memória de restrição da extensão de $\mathbf{w} = cc$ por c, ou seja, $\mathcal{R}(ccc)$. Como $(c, \square) \in \mathcal{M}(cc)$ e $ccc \notin \mathcal{P}(\mathcal{OA}^{-1})$, é necessário determinar a memória de restrição do sufixo próprio mais longo de cc em $\mathcal{P}(\mathcal{OA}^{-1})$ estendido por c (pelo caso 4). Da Figura 3.1.1, o sufixo próprio mais longo de ccc em $\mathcal{P}(\mathcal{OA}^{-1})$ é cc. Assim $\mathcal{R}(ccc) = \mathcal{R}(cc)$, no entanto, se as memórias de restrição das palavras em $\mathcal{P}(\mathcal{OA}^{-1})$ forem determinadas de palavras mais curtas para palavras mais longas, $\mathcal{R}(cc)$ já foi determinado. Observe que $cc \in \mathcal{P}(\mathcal{OA}^{-1})$, logo $\mathcal{R}(cc) = cc$, pelo Caso 3).

4 GRAFO DE CONTEXTOS

4.1 Identificação de Palavras Equivalentes

Esta seção introduz um algoritmo do tipo Moore que cria sucessivas partições de $\mathcal{P}(\mathcal{OA}^{-1})$ em classes de palavras equivalentes (com o mesmo conjunto de contextos). Este algoritmo é baseado nos conceitos de máscara de restrição e memória de restrição, com o primeiro definindo a partição inicial enquanto a etapa recursiva faz uso da memória de restrição. Quando o critério de parada é satisfeito, as classes de palavras equivalentes em $\mathcal{P}(\mathcal{OA}^{-1})$ são obtidas.

As partições definidas a seguir são empregadas no algoritmo apresentado em Chaves (2014).

Definição 4.1 (Partição por Máscara de Restrição)**.** A partição de $\mathcal{P}(\mathcal{OA}^{-1})$ em relação à $a \in \mathcal{A}$, denotada por $a|\mathcal{P}(\mathcal{OA}^{-1})$, resulta em no máximo três conjuntos determinados como:

$$\begin{cases} P_1 = \{\mathbf{w} \in \mathcal{P}(\mathcal{OA}^{-1}) | \ \ (a,\ \blacksquare) \in \mathcal{M}(\mathbf{w})\}, \\ \\ P_2 = \{\mathbf{w} \in \mathcal{P}(\mathcal{OA}^{-1}) | \ \ (a,\ \square) \in \mathcal{M}(w)\}, \\ \\ P_3 = \mathcal{P}(\mathcal{OA}^{-1}) \backslash (P_1 \cup P_2). \end{cases}$$

Definição 4.2 (Partição por Memória de Restrição)**.** Seja $P \subseteq \mathcal{P}(\mathcal{OA}^{-1})$ e $a \in \mathcal{A}$. Uma seção de $\mathcal{P}(\mathcal{OA}^{-1})a$ com respeito à P é o conjunto:

$$P \backslash \mathcal{P}(\mathcal{OA}^{-1})a = \{\mathbf{w} \in P(\mathcal{OA}^{-1}) | \quad \mathbf{w}a \in L \text{ e } \mathcal{R}(\mathbf{w}a) \in P\}.$$

Especifica-se $(P,\ a)|\mathcal{P}(\mathcal{OA}^{-1}) = \{P_1,\ P_1^c\}$ como a partição de $\mathcal{P}(\mathcal{OA}^{-1})$ em relação à P e o símbolo $a \in \mathcal{A}$, onde $P_1 = P \backslash \mathcal{P}(\mathcal{OA}^{-1})a$ e P_1^c é o complemento de P_1 relativo à $\mathcal{P}(\mathcal{OA}^{-1})$.

O algorítmo é apresentado na Tabela 4.1 e é dividido em três passos, descritos a seguir.

Passo 1. Particionamento por máscara de restrição: a partição inicial $\mathcal{P}^1 = \{P_1, P_2, \ldots, P_n\}$ é a partição mais densa de $\mathcal{P}(\mathcal{OA}^{-1})$ em relação à $a|\mathcal{P}(\mathcal{OA}^{-1})$ (ver Definição 4.1) para todo $a \in \mathcal{A}$.

Passo 2. Particionamento por memória de restrição: neste passo recursivo, o particionamento proposto na Definição 4.2 é empregado para refinar a partição anterior $\mathcal{P}^k, k \geq 1$. Deve-se notar que as palavras em $P \in \mathcal{P}^k$ são apenas estendidas por um símbolo $a \in \mathcal{A}$ se $(a, \square) \in \mathcal{M}(P)$. Esse passo gera a partição mais densa de

Tabela 4.1 – *Algoritmo Proposto para o SFT.*

1. $\mathcal{P}^1 \leftarrow \wedge_{a\in\mathcal{A}}\ a\|\mathcal{P}(\mathcal{OA}^{-1})$	▷ A partição inicial (Passo 1)
2. $k \leftarrow 0$	
3. repetir	▷ Iniciar o passo 2
4. $k \leftarrow k+1$	
5. para todo $a \in \mathcal{A}$ fazer	
6. $\mathcal{P}_a \leftarrow \wedge_{P\in\mathcal{P}^k}\ (P,a)\,\|\mathcal{P}(\mathcal{OA}^{-1})$	
7. $\mathcal{P}^{k+1} \leftarrow \mathcal{P}^k \wedge \bigwedge_{a\in\mathcal{A}}\ \mathcal{P}_a$	▷ Nova Partição
8. até $\mathcal{P}^{k+1} = \mathcal{P}^k$	▷ Passo 3

Fonte: Chaves (2014).

$\mathcal{P}(\mathcal{OA}^{-1})$ relativa a todo $P \in \mathcal{P}^k$ pela extensão de um símbolo específico $a \in \mathcal{A}$ (Linha 6 na Tabela 4.1) e na Linha 7 as partições resultantes deste processo (para cada $a \in \mathcal{A}$) refina $\mathcal{P}^k$.

Passo 3. Critério de Parada: O algoritmo termina quando refinamentos não são mais possíveis.

A consistência do algoritmo é provada a seguir.

Lema 4.1. *Em cada partição $\mathcal{P}^k$, as palavras em $\mathcal{P}(\mathcal{OA}^{-1})$ em conjuntos (ou classes) distintos têm conjuntos de contextos distintos.*

Demonstração. A prova é por indução. A afirmação é verdadeira para a primeira partição $\mathcal{P}^1$ (obtida no Passo 1), pois, para quaisquer palavras $\mathbf{u}, \mathbf{v} \in \mathcal{P}(\mathcal{OA}^{-1})$ se $\mathcal{M}(\mathbf{u}) \neq \mathcal{M}(\mathbf{v})$, então existe uma palavra em $F(\mathbf{u})$ que não pertence a $F(\mathbf{v})$, ou vice-versa, o que implica que $F(\mathbf{u}) \neq F(\mathbf{v})$. Agora suponha que a afirmação é verdadeira para $\mathcal{P}^k,\ k \geq 2$. Sejam $\mathbf{v},\ \mathbf{w} \in \mathcal{P}(\mathcal{OA}^{-1})$ palavras na mesma classe em $\mathcal{P}^k$, mas em classes distintas em $\mathcal{P}^{k+1}$. A partir do Passo 2, isso ocorre porque existe um $a \in \mathcal{A}$ tal que $\mathcal{R}(\mathbf{v}a)$ e $\mathcal{R}(\mathbf{w}a)$ estão em classes distintas em $\mathcal{P}^k$. Assim, da hipótese indutiva, $\mathcal{R}(\mathbf{v}a)$ e $\mathcal{R}(\mathbf{w}a)$ possuem conjuntos de contextos distintos, portanto $\mathbf{v}$ e $\mathbf{w}$ também possuem conjuntos de contextos distintos. Logo, o lema é válido para $\mathcal{P}^{k+1}$. □

Teorema 4.1. *O algoritmo proposto particiona $\mathcal{P}(\mathcal{OA}^{-1})$ em classes de palavras equivalentes (com o mesmo conjunto de contextos).*

Demonstração. Com o resultado do Lema 4, precisa-se provar que a condição de parada é alcançada quando palavras na mesma classe possuem o mesmo conjunto de contextos. Suponha que as partições $\mathcal{P}^k$ e $\mathcal{P}^{k+1}$ sejam iguais, mas existam $\mathbf{v},\ \mathbf{w} \in \mathcal{P}(\mathcal{OA}^{-1})$ na mesma classe com conjuntos de contextos distintos. Agora, a prova segue por contradição. A memória finita do SFT implica que existe uma palavra $\mathbf{v} \in L,\ \ \mathbf{v} = v_1 \dots v_n$, tal que $\mathbf{wv},\ \mathbf{uv} \in L$, mas, $\mathbf{wv}a \in L$ e $\mathbf{uv}a \notin L$ para algum $a \in \mathcal{A}$. Portanto, segue-se que $(a,\ \blacksquare) \notin \mathcal{M}(\mathcal{R}(\mathbf{wv}))$ e $(a,\ \blacksquare) \in \mathcal{M}(\mathcal{R}(\mathbf{uv}))$, daí as memórias de restrição $\mathcal{R}(\mathbf{wv})$ e $\mathcal{R}(\mathbf{uv})$ devem pertencer a classes distintas em $\mathcal{P}^1$.

Como $\mathcal{P}^k$ e $\mathcal{P}^{k+1}$ são iguais, então $\mathcal{R}(\mathbf{w}v_1)$ e $\mathcal{R}(\mathbf{u}v_1)$ estão na mesma classe em $\mathcal{P}^k$. Como consequência, $\mathcal{R}(\mathbf{w}v_1v_2)$ e $\mathcal{R}(\mathbf{u}v_1v_2)$ estão na mesma classe em $\mathcal{P}^{k-1}$. Em geral, $\mathcal{R}(\mathbf{w}v_1 \dots v_i)$ e $\mathcal{R}(\mathbf{u}v_1 \dots v_i)$ estão na mesma classe em $\mathcal{P}^{k-i+1}$. Finalmente, $\mathcal{R}(\mathbf{wv})$ e $\mathcal{R}(\mathbf{uv})$ devem pertencer a mesma classe em $\mathcal{P}^{k-n+1}$. De fato, a última afirmação é uma contradição, desde que $\mathcal{R}(\mathbf{wv})$ e $\mathcal{R}(\mathbf{uv})$ estão em classes distintas em $\mathcal{P}^1$. □

Exemplo 4.1. Vamos utilizar o algoritmo da Tabela 4.1 para particionar o conjunto $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$ do Exemplo 3.1 em classes de palavras com o mesmo contexto à direita. Seguindo o Passo 1 do algoritmo, os conjuntos $\mathcal{M}(\mathbf{w})$ serão calculados para todos os $\mathbf{w} \in \mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$. Como no Exemplo 3.2 já foram calculadas $\mathcal{M}(a)$, $\mathcal{M}(c)$, $\mathcal{M}(cc)$ e $\mathcal{M}(acc)$, então basta calcular as máscaras de restrição das demais palavras.

1) Para $\mathbf{w} = b$, temos:

$$b^{-1}\mathcal{O} = \{cdd,\ bdd\} \ \Rightarrow\ (b,\ \square),\ (c,\ \square)\ \in\ \mathcal{M}(b).$$

Como o sufixo próprio mais longo de b em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$ é ε e $\mathcal{M}(\varepsilon) = \varnothing$ então,

$$\mathcal{M}(b)\ =\ \{(b,\ \square),\ (c,\ \square)\}.$$

2) Para $\mathbf{w} = ab$, temos:

$$(ab)^{-1}\mathcal{O} = \{c\}\ \Rightarrow\ (c,\ \blacksquare)\ \in\ \mathcal{M}(ab).$$

Como o sufixo próprio mais longo de ab em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$ é b e $\mathcal{M}(b)\ =\ \{(b,\ \square),\ (c,\ \square)\}$, então, deve-se acrescentar $(b,\ \square)$ à $\mathcal{M}(ab)$. Daí,

$$\mathcal{M}(ab)\ =\ \{(b,\ \square),\ (c,\ \blacksquare)\}.$$

3) Para $\mathbf{w} = ac$, temos:

$$(ac)^{-1}\mathcal{O} = \{d,\ cb\}\ \Rightarrow\ (c,\ \square),\ (d,\ \blacksquare)\ \in\ \mathcal{M}(ac).$$

Como o sufixo próprio mais longo de ac em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$ é c e $\mathcal{M}(c)\ =\ \{(c,\ \square)\}$, assim,

$$\mathcal{M}(ac)\ =\ \{(c,\ \square),\ (d,\ \blacksquare)\}.$$

4) Para $\mathbf{w} = bb$, temos:

$$(bb)^{-1}\mathcal{O} = \{dd\}\ \Rightarrow\ (d,\ \square)\ \in\ \mathcal{M}(bb).$$

Como o sufixo próprio mais longo de bb em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$ é b e $\mathcal{M}(b)\ =\ \{(b,\ \square),\ (c,\ \square)\}$

então, devemos acrescentar os pares $(b,\ \square)$, $(c,\ \square)$ em $\mathcal{M}(bb)$. Daí,

$$\mathcal{M}(bb) \ = \ \{(b,\ \square),\ (c,\ \square),\ (d,\ \square)\}.$$

5) Para $\mathbf{w} = bc$, temos:

$$(bc)^{-1}\mathcal{O} = \{dd\} \ \Rightarrow \ (d,\ \square) \ \in \ \mathcal{M}(bc).$$

Como o sufixo próprio mais longo de bc em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$ é c e $\mathcal{M}(c) \ = \ \{(c,\ \square)\}$, logo devemos acrescentar o par $(c,\ \square)$ à $\mathcal{M}(bc)$. Portanto,

$$\mathcal{M}(bc) \ = \ \{(c,\ \square),\ (d,\ \square)\}.$$

6) Para $\mathbf{w} = bcd$, temos:

$$(bcd)^{-1}\mathcal{O} = \{d\} \ \Rightarrow \ (d,\ \blacksquare) \ \in \ \mathcal{M}(bcd).$$

Como o sufixo próprio mais longo de bcd em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$ é ε e $\mathcal{M}(\varepsilon) = \varnothing$, então

$$\mathcal{M}(bcd) \ = \ \{(d,\ \blacksquare)\}.$$

7) Para $\mathbf{w} = ccd$, temos:

$$(ccd)^{-1}\mathcal{O} = \{b\} \ \Rightarrow \ (b,\ \blacksquare) \ \in \ \mathcal{M}(ccd).$$

Como o sufixo próprio mais longo de ccd em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$ é ε e $\mathcal{M}(\varepsilon) = \varnothing$,por isso ,

$$\mathcal{M}(ccd) \ = \ \{(b,\ \blacksquare)\}.$$

8) Para $\mathbf{w} = bbd$, temos:

$$(bbd)^{-1}\mathcal{O} = \{d\} \ \Rightarrow \ (d,\ \blacksquare) \ \in \ \mathcal{M}(bbd).$$

Como o sufixo próprio mais longo de bbd em $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$ é ε e $\mathcal{M}(\varepsilon) = \varnothing$, então,

$$\mathcal{M}(bbd) \ = \ \{(d,\ \blacksquare)\}.$$

As respectivas máscaras de restrição estão listadas na Tabela 4.2.

De acordo com o Passo 1, a partição inicial é

$$\mathcal{P}^1 = \{\{\varepsilon\},\ \{a,\ b\},\ \{c\},\ \{ab\},\ \{ac\},\ \{cc,\ bc\},\ \{bb\},\ \{acc\},\{ccd\},\ \{bcd,\ bbd\}\}.$$

Tabela 4.2 – *Máscaras de Restrição dos Elementos de $\mathcal{P}(\mathcal{OA}^{-1})$ do Exemplo 3.1.*

$\mathbf{w} \in \mathcal{P}(\mathcal{OA}^{-1})$	$\mathcal{M}(\mathbf{w})$
ε	$\varnothing$
a	$\{(b,\ \square),\ (c,\ \square)\}$
b	$\{(b,\ \square), (c,\ \square)\}$
c	$\{(c,\ \square)\}$
ab	$\{(b,\ \square),\ (c,\ \blacksquare)\}$
ac	$\{(c,\ \square),\ (d,\ \blacksquare)\}$
bb	$\{(b,\ \square),\ (c,\ \square),\ (d,\ \square)\}$
bc	$\{(c,\square),\ (d,\square)\}$
cc	$\{(c,\square),\ (d,\square)\}$
acc	$\{(b,\ \blacksquare),\ (c,\ \square),\ (d,\ \square)\}$
bbd	$\{(d,\ \blacksquare)\}$
bcd	$\{(d,\ \blacksquare)\}$
ccd	$\{(b,\ \blacksquare)\}$

Fonte: Elaborado pelo autor, 2022.

Agora faça

$$\begin{cases} P_1 = \{\varepsilon\}, & P_2 = \{a,\ b\}, & P_3 = \{c\}, & P_4 = \{ab\}, & P_5 = \{ac\}, \\ P_6 = \{cc,\ bc\}, & P_7 = \{bb\}, & P_8 = \{acc\}, & P_9 = \{ccd\} & \text{e } P_{10} = \{bcd,\ bbd\} \end{cases}.$$

Uma vez que os conjuntos unitários de qualquer $\mathcal{P}^k$ não precisam ser considerados na Etapa 2, então só se faz necessário analisar os subconjuntos $P_2 = \{a,\ b\}$, $P_6 = \{cc,\ bc\}$ e $P_{10} = \{bcd,\ bbd\}$. Além disso, as extensões a serem consideradas estão associadas apenas à marca $\square$. Os elementos de P_2 em $\mathcal{P}^1$ podem ser estendidos pelo símbolo b, resultando no conjunto $\{ab,\ bb\}$ e,

$$\begin{cases} \mathcal{R}(ab) = ab \in P_4 \\ \qquad\quad \text{e} \qquad\quad . \\ \mathcal{R}(bb) = bb \in P_7 \end{cases}$$

Como as memórias de restrição das extensões destas palavras estão em conjuntos distintos de $\mathcal{P}^1$, logo, elas devem ser separadas. Além disso, os elementos de P_6 *em* $\mathcal{P}^1$ podem ser estendidos pelo símbolo d, resultando no conjunto $\{ccd,\ bcd\}$ e,

$$\begin{cases} \mathcal{R}(ccd) = ccd \in P_9 \\ \qquad\quad \text{e} \qquad\quad . \\ \mathcal{R}(bcd) = bcd \in P_{10} \end{cases}$$

De forma análoga, as memórias de restrição das extensões das palavras cc e bc pelo símbolo d estão em conjuntos distintos de $\mathcal{P}^1$, devendo também serem separadas. Daí,

tem-se o refinamento

$$\mathcal{P}^2 = \{\{\varepsilon\},\ \{a\},\ \{b\},\ \{c\},\ \{ab\},\ \{ac\},\ \{cc\},\ \{bc\},\ \{bb\},\ \{acc\},\ \{ccd\},\ \{bcd,\ bbd\}\},$$

em que

$$\begin{cases} P_1 = \{\varepsilon\}, & P_2 = \{a\}, & P_3 = \{b\}, & P_4 = \{c\}, & P_5 = \{ab\}, & P_6 = \{ac\}, \\ P_7 = \{cc\}, & P_8 = \{bc\}, & P_9 = \{bb\}, & P_{10} = \{acc\}, & P_{11} = \{ccd\} & \text{e } P_{12} = \{bcd,\ bbd\} \end{cases}.$$

Na segunda iteração do Passo 2, a extensão de qualquer elemento em $P_{12} = \{bcd,\ bbd\}$ tem memória de restrição o próprio elemento. Essa partição não pode ser refinada e o critério de parada no Passo 3 é alcançado.

4.2 Construção do Grafo de Contextos

Esta seção apresenta a construção do grafo de contextos de um SFT. A descrição dessa construção é dada em Lind (1995), sendo o grafo $G = (\mathcal{V}, \mathcal{E}, \mathcal{L})$ tal que

$$\mathcal{V} \ = \ \{F(\mathbf{w}) |\ w \in L\}$$

e existe uma aresta $e \in \mathcal{E}$, com $\mathcal{L}(e) = a$ do vértice $F(\mathbf{w})$ para o vértice $F(\mathbf{w}')$ se, e somente se, $F(\mathbf{w}a) = F(\mathbf{w}')$.

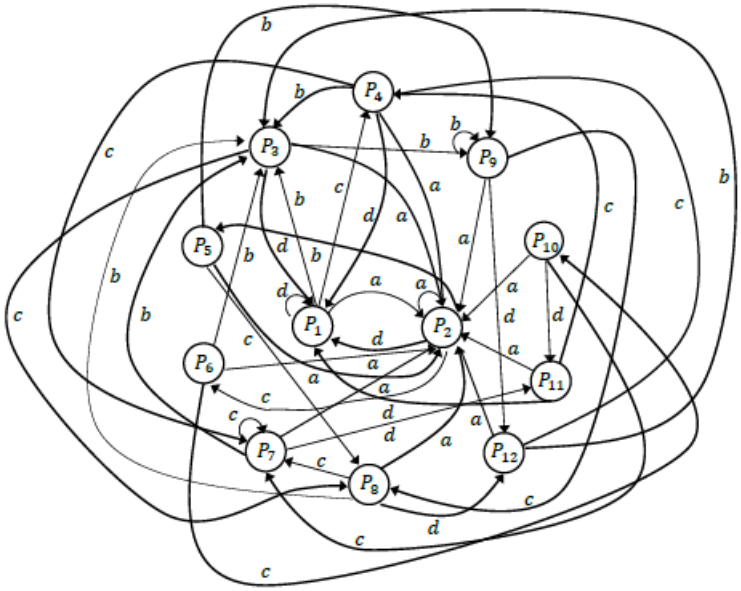
A partir da correspondência entre conjuntos de contextos e conjuntos de restrições (Teorema 3.1), o conjunto de vértices $\mathcal{V}$ pode ser construído como

$$\mathcal{V} \ = \ \{\mathcal{C}(\mathbf{w}) |\ \mathbf{w} \in \mathcal{P}(\mathcal{O}\mathcal{A}^{-1})\},$$

sendo, portanto, os elementos da partição alcançados no Passo 3 do algoritmo descrito na Tabela 4.1.

Para determinar as transições de vértices aplicamos a Proposição 3.1, assim, existe uma aresta rotulada como a de uma classe P_i para P_j se $\mathbf{w} \in P_i$, $\mathcal{R}(\mathbf{w}a) \in P_j$ e $(a,\ \blacksquare) \notin \mathcal{M}(\mathbf{w})$. Para um símbolo $a \in \mathcal{A}$ pertecendo a $\mathcal{M}(\mathbf{w})$ com bandeira $\square$ a memória de restrição $\mathcal{R}(\mathbf{w}a)$ já foi calculada na Etapa 2 do processo de partição. Observe que $\mathcal{R}(\mathbf{w}a) = \mathcal{R}(a)$ para qualquer a não pertencente à $\mathcal{M}(\mathbf{w})$, então o vértice terminal da aresta correspondente deve conter $\mathcal{R}(a)$. Seguindo esse processo, é possível reproduzir a Figura 4.2.1.

Figura 4.2.1 – *Grafo de Contextos gerado a partir da partição* $\mathcal{P}^2$ *de* $\mathcal{P}(\mathcal{OA}^{-1})$ *obtida no Exemplo 4.1.*



Fonte: Elaborado pelo autor, 2022.

5 CONSIDERAÇÕES FINAIS

Em Engenharia Elétrica é frequente, no estudo de sinais, o aparecimento de sequências com restrições. Para interpretar tais sinais são necessários codificadores restritivos e na construção desses codificadores são utilizados algorítmos que permitam reconhecer pontos de sistemas simbólicos invariantes por deslocamento, em particular, dos SFT's. Os algorítmos, em geral, são tão mais eficientes quanto menor for o tempo de execução da tarefa para o qual ele foi proposto. Nessa linha de raciocínio, o nosso trabalho fez um estudo minucioso do algorítmo que gera os estados da apresentação mínima de um SFT, bem como mostrou a construção do grafo de contextos, ou seja, a representação geométrica do SFT.

A complexidade da construção da partição inicial do algorítmo proposto é comparável ao alcançado por alguns métodos propostos na literatura para encontrar uma apresentação determinística inicial de um SFT. Alguns desses métodos são baseados na árvore do conjunto de palavras proibidas mínimas e tem complexidade linear no tamanho n da árvore, em que n é a cardinalidade de $\mathcal{P}(\mathcal{O}\mathcal{A}^{-1})$. Uma característica distinta do procedimento descrito no nosso trabalho é que as transições de vértices são calculadas após a determinação das palavras equivalentes. Esse conjunto pode ser muito menor que o conjunto de vértices de uma apresentação inicial, reduzindo, assim, a complexidade na construção do grafo de contextos.

REFERÊNCIAS

BERSTEL, J.; BOASSON, L.; CARTON, O.; FAGNOT, I. **Minimization of automata**. *CoRR*, vol. abs/1010.5318, 2010.

CHAVES, D. P. B.; PIMENTEL, C. **On the Follower Set Graph of Shifts of Finite Type**. *International Symposium on Information Theory*, (Honolulu, HI, USA), pp. 1802-1806, 2014.

CHAVES, D. P. B.; PIMENTEL, C. **A new algorithm for finding the shannon cover of shifts of finite type**. *in Proc. International Telecommunications Symposium*, (Fortaleza, CE, Brazil), pp. 694-699, 2006.

CROCHEMORE, M.; MIGNOSI, F.; RESTIVO, A. **Automata and forbidden words**. *Inform. Process. Lett.*, vol. 67, pp. 111-117, 1998.

DUMMIT, D.; FOOTE, R. **Abstract Algebra. Wiley**, 2004.

HOPCROFT, J. E. **An $(n \cdot log\ n)$ algorithm for minimizing states in a finite automaton**. Tech. Rep. STAN-CS-71-190, Stanford University, 1971.

IMMINK, K. A. S. **Efmplus: The coding format of the multimedia compact disc**. *IEEE Trans. Consum. Electron.*, vol. 41, no. 3, pp. 491-497, 1995.

LIND, D.; MARCUS, B. H. **An Introduction to Symbolic Dynamics and Coding**. Cambridge University Press, 1995.

MANADA, A.; KASHYAP, N. **On the shannon covers of certain irreducible constrained systems of finite type**. *2006 IEEE International Symposium on Information Theory*, pp. 1477-1481, July 2006.

MARCUS, B. H.; ROTH, R. M.; SIEGEL, P. H. **Constrained systems and coding for recording channels**. *Handbook of Coding Theory* (V. S. Pless and W. Huffman, eds.), vol. 2, pp. 1635-1764, Eds. Amsterdam:Elsevier, 1999.

OLIVEIRA, V. A.; RANGEL, S.; ARAUJO, S. A. **Teoria dos Grafos**. ibilce.unesp.br.