



Departamento de Engenharias e Ciência da Computação  
Curso de Ciência da Computação  
Projeto Integrador II – Algoritmo de AHO-CORASICK

## **BUSCA DE PALAVRAS**

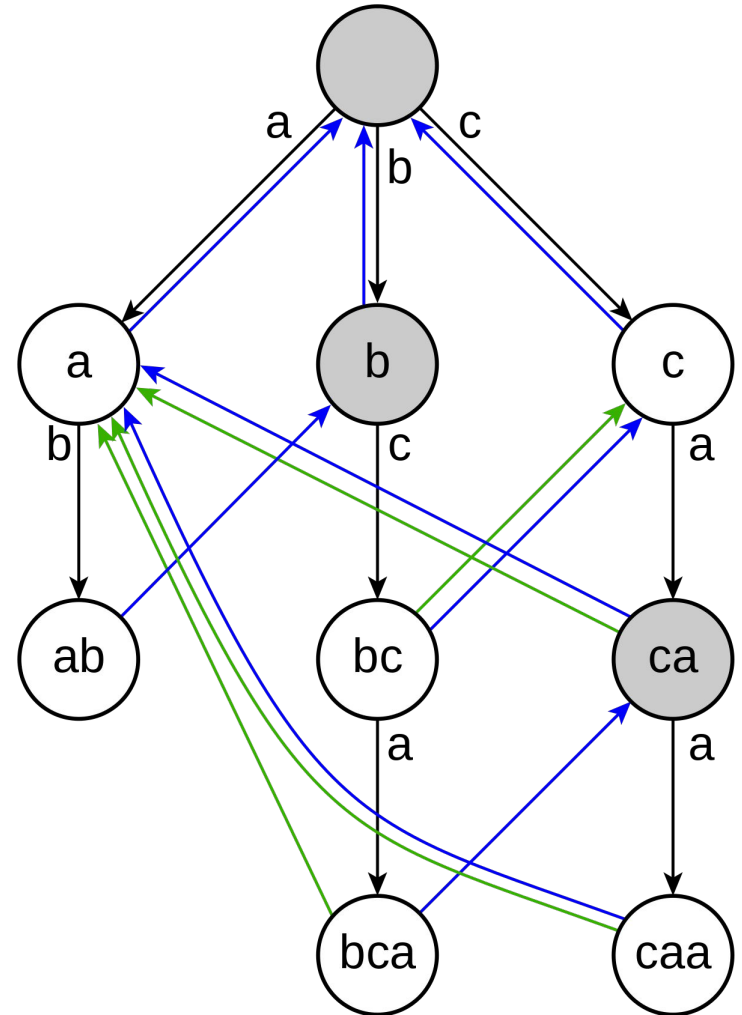
Alderí Pasa Júnior, João Rosseto,  
Gabriel Barp Deon e Otávio Luiz Simgame Piana.

Erechim, 2022

A principal motivação para a realização do projeto e desenvolvimento do aplicativo, foi a vontade de criar uma ferramenta simples que auxilia usuários na aprendizagem de outros idiomas, ajudando principalmente na hora de encontrar palavras-chave dentro de textos de assuntos de seu interesse.

## Aho-Corasik

O algoritmo Aho-Corasick é amplamente utilizado em diversas aplicações, como processamento de linguagem natural, análise de logs de sistemas, detecção de vírus em arquivos, entre outras. Sua eficiência e simplicidade tornam-no uma das principais escolhas para a busca de múltiplas cadeias de caracteres em um texto.



## Delphi

"O Delphi é uma linguagem de programação orientada a objetos e um ambiente de desenvolvimento integrado (IDE) para desenvolvimento rápido de aplicativos (RAD), com uma grande biblioteca de componentes de interface do usuário prontos para uso" (Jenkins & Powell, 2002, p. 70).



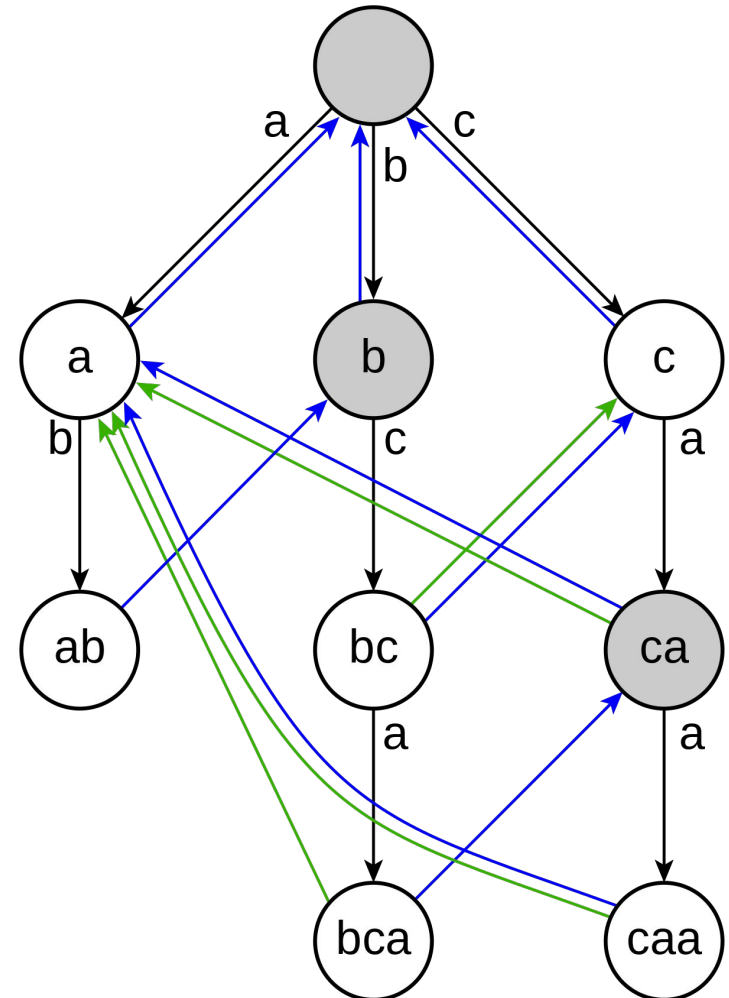
## Libre Translate - API

A API LibreTranslate é uma ferramenta de tradução de código aberto que permite a tradução automática de texto entre vários idiomas. Com base em modelos de tradução neural avançados, a API fornece uma interface fácil de usar para integrar a funcionalidade de tradução em aplicativos, sites e serviços. Ela oferece suporte a uma ampla gama de idiomas e possui recursos adicionais, como detecção de idioma e transliteração.



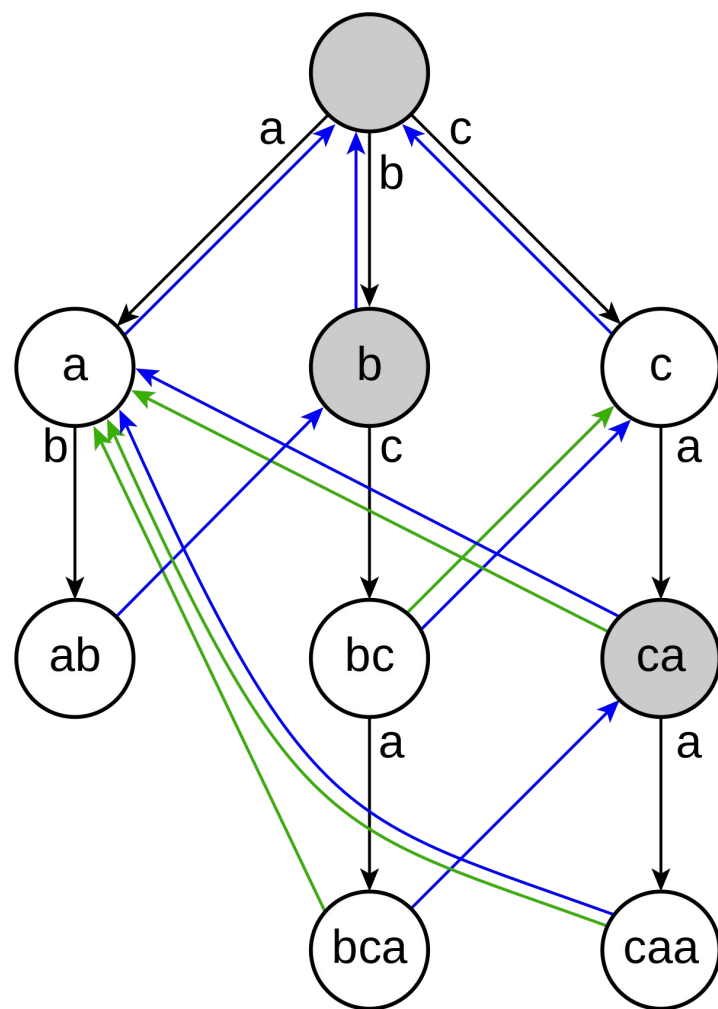
## Construção da árvore de prefixos

Os padrões de busca são inseridos em uma árvore de prefixos, onde cada nó representa um prefixo de um padrão. As arestas são rotuladas com os caracteres que podem ser seguidos a partir de um nó específico.



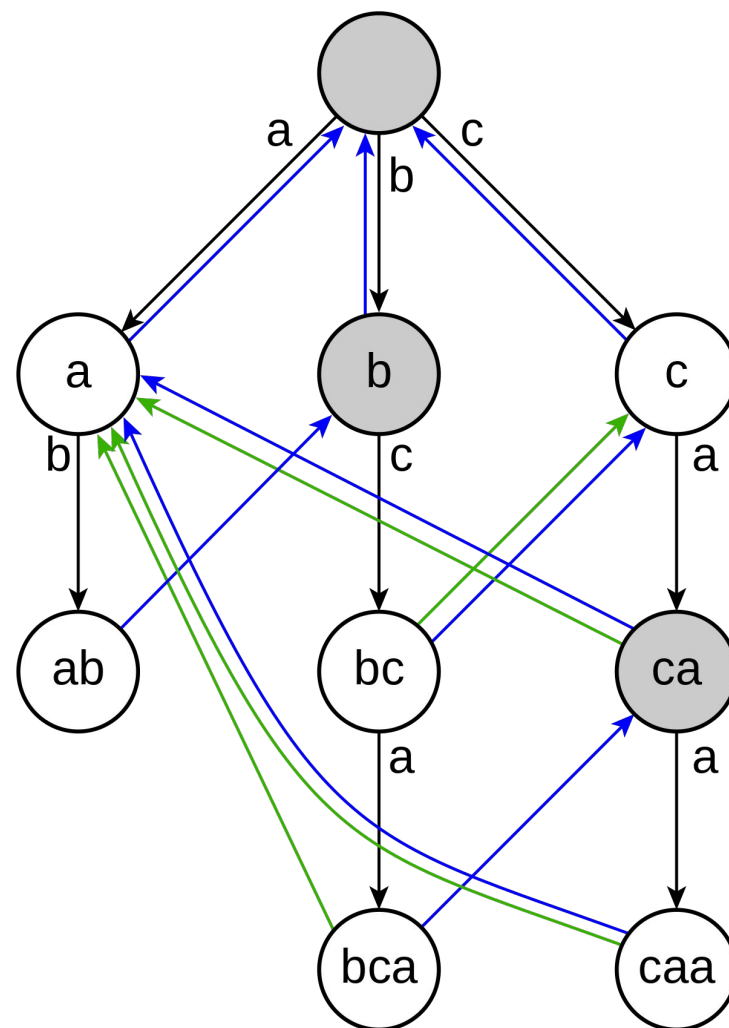
## Falhas

Para otimizar a busca, são adicionadas transições de falha na árvore de prefixos. Cada nó na árvore, exceto a raiz, possui uma transição de falha que aponta para o nó correspondente ao maior sufixo próprio desse nó que também está presente na árvore. Isso permite que o algoritmo continue a busca em outros padrões, mesmo quando um padrão atual não tem correspondência.



## Busca de padrões

Para realizar a busca, o texto de entrada é processado caractere por caractere. O estado atual do algoritmo é representado pelo nó atual na árvore de prefixos. O algoritmo segue as transições na árvore de acordo com os caracteres do texto de entrada. Se uma correspondência completa for encontrada, um padrão foi localizado e um registro é feito.





# **DEMONSTRAÇÃO PRÁTICA**

## Código

**begin**

```
AhoCorasick := TAhoCorasick.Create;
```

```
mmRichEdit.SelectAll;
```

```
mmRichEdit.SelAttributes := mmRichEdit.DefAttributes;
```

**try**

*// Adiciona as palavras chaves*

```
AdicionarPalavraPesquisa(edtPalavra1, AhoCorasick);
```

```
AdicionarPalavraPesquisa(edtPalavra2, AhoCorasick);
```

```
AdicionarPalavraPesquisa(edtPalavra3, AhoCorasick);
```

*//Adiciona as Traduções*

```
AdicionarPalavraPesquisa(edtTraducao1, AhoCorasick);
```

```
AdicionarPalavraPesquisa(edtTraducao2, AhoCorasick);
```

```
AdicionarPalavraPesquisa(edtTraducao3, AhoCorasick);
```

*//Constroi a estrutura de busca*

```
AhoCorasick.Build;
```

```
//Realize a busca no texto
```

```
if ckbCaseSensitive.Checked then
```

```
begin
```

```
Matches := AhoCorasick.FindWords(mmRichEdit.Text);
```

```
end else
```

```
begin
```

```
Matches := AhoCorasick.FindWords(LowerCase(mmRichEdit.Text));
```

```
end;
```

```
//Exibe as palavras encontradas no log
```

```
if Length(Matches) > 0 then
```

```
begin
```

```
mmLog.Clear;
```

```
pnLog.Visible := True;
```

```
for i := 0 to Length(Matches) - 1 do
```

```
begin
```

```
if ckbDestacar.Checked then
```

```
begin
```

```
FindAndHighlightWord(mmRichEdit, Matches[i].Word);
```

```
end;
```

# CONSIDERAÇÕES FINAIS

Ocurrence Finder

Configurações

Palavra Chave

Tradução para

Nenhum

Palavra Chave

Samurai

Tradução para

Nenhum

Palavra Chave

honor

Tradução para

Português

honra.

☒ Destacar Ocorrências

☒ Case Sensitive

Procurar

Log

--Palavra encontrada: "samurai" (Posição: 4)

--Palavra encontrada: "samurai" (Posição: 259)

--Palavra encontrada: "samurai" (Posição: 631)

--Palavra encontrada: "samurai" (Posição: 857)

Texto Base

Os **samurais**, também conhecidos como bushis, eram guerreiros da classe militar no Japão feudal. Eles eram considerados a elite militar e tinham um código de conduta rigoroso chamado bushido, que valorizava a lealdade, a honra, a coragem e a disciplina.

Os **samurais** eram habilidosos no combate, treinando intensamente em artes marciais, como o kenjutsu (arte da espada), o kyujutsu (arco e flecha) e o naginatajutsu (luta com naginata, uma lança de lâmina longa). Eles dominavam técnicas de luta tanto a cavalo quanto a pé, e seu objetivo era se tornarem guerreiros eficientes e mortais.

Além de sua habilidade em combate, os **samurais** também valorizavam a educação e a cultura. Muitos eram letrados, estudando literatura, poesia, filosofia e estratégias militares. Eles buscavam o equilíbrio entre a força física e o desenvolvimento intelectual.

Os **samurais** serviam a um senhor feudal, conhecido como daimyo, e eram responsáveis por proteger suas terras e honrar sua lealdade. Eles viviam por um código rigoroso de ética e honra, e o desonrar do bushido poderia resultar em seppuku, um suicídio ritual para preservar a dignidade.

Limpar

"Text Algorithms" escrito por Maxime Crochemore e Wojciech Rytter em 1994

"Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology" - Dan Gusfield.

Boyer, R. S., & Moore, J. S. (1977). A fast string searching algorithm. Communications of the ACM.

Aho, A. V., & Corasick, M. J. (1975). Efficient string matching: an aid to bibliographic search. Communications of the ACM,

Navarro, G. (2001). A guided tour to approximate string matching. ACM Computing Surveys (CSUR).

**(algoritmo de aho-corasick)**

Jenkins, A., & Powell, K. (2002). Beginning Delphi 7. Apress. **(Delphi)**

Knuth, D. E., Morris, Jr, J. H., & Pratt, V. R. (1977). Fast pattern matching in strings. SIAM Journal on Computing. **(algoritmo de knuth-morris-pratt)**

Obrigado pela atenção!