Universidade Federal de Mato Grosso do Sul (UFMS)
Faculdade de Computação

Rorschach: Uma Ferramenta para Detecção de Plágio

Yuri Karan Benevides Tomas, Edna Ayako Hoshino (orientadora)

12 de dezembro de 2014

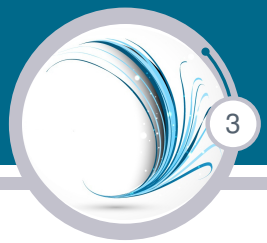
Introdução
Plágio

Plagiar é o ato de assumir autoria ou utilizar como fonte uma obra intelectual pertencente a outra pessoa.

- Textos;
- Músicas;
- Filmes; e
- Códigos-fonte.

Ferramentas que utilizam o RKR-GST

- CPD
- JPlag
- Marble
- Plaggie (GNU GPL)
- YAP3

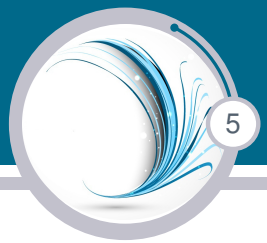


Rorschach utiliza a licença GNU GPL versão 3

- Uso;
- Modificação; e
- Compartilhamento;

Metodologia
Alfabeto e *letra*

- *Alfabeto* - conjunto finito e não vazio de elementos.
- Estes elementos são chamados de *letras*.
- Σ representa um *alfabeto* arbitrário.



Exemplos de *alfabeto*

- $\Sigma = \{0, 1\}$, o alfabeto binário;
- $\Sigma = \{0, 1, ..., 9\}$, o alfabeto numérico;
- $\Sigma = \{a, b, ..., z\}$, o alfabeto das letras minúsculas; e
- O conjunto de caracteres que compõem o código ASCII.

Metodologia
Palavra

- Sequência de *letras*.
- Sendo uma *palavra w*, sobre um alfabeto Σ, temos que:
  - w_i é a i-ésima letra de *w*.
  - $|w|$ é o comprimento, ou tamanho, de *w*.

Metodologia
Subcadeia

- Uma *subcadeia* de *w* é uma *palavra x* cujas *letras* pertencem a *w* e estão em *x* na mesma sequência que em *w*.
- *x* é *subcadeia* da *palavra y* se $\exists$ *z* e *w*, *subcadeias*, tal que *zxw* = *y*.

Exemplos:

- metodologia
- metodologia
- metodologia

Metodologia
Função *Hash*

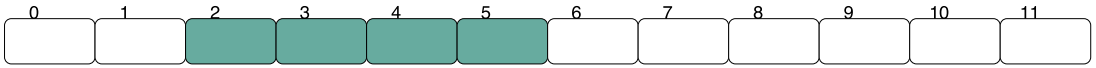
- *Função hash* - Mapeamento de informação sem tamanho fixo para uma de tamanho fixo.
- valor de *hash* - o valor retornado pela função de *hash*.

Exemplo

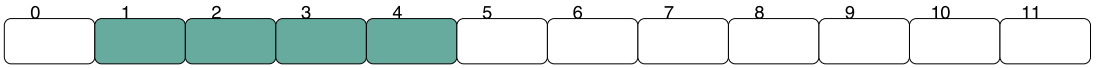
- Mapeamento de números inteiros para o intervalo de inteiros $[0, 99]$.
- Função de *hash* $f(x) = x \mod 100$.
- Assim o valor de *hash* para o número 1039 seria 39.

Metodologia
Janela

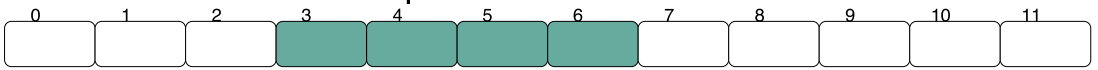
Janela A de tamanho 4, começando na posição 2 da palavra:



Janela A após deslocamento à esquerda



Janela A após deslocamento à direita



Janela A após dois deslocamentos à esquerda

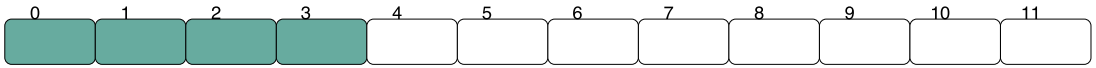


Figura: Exemplo de janela.

Metodologia
Algoritmo Rolling Hash

- Cada *palavra* é associada a um número em uma base numérica especial definida pelo tamanho do alfabeto.
- Cada *letra* do *alfabeto* é associada a um algarismo do número.
- Assim, sendo w uma palavra, $hash(w) = \sum_{i=1}^{|w|} w_i * base^{|w|-i}$

Exemplo:

Índice da posição	3	2	1
Código ASCII extendido	115	111	109
Caractere	s	o	m

Tabela: Valores correspondentes na tabela ASCII extendida.

Valor de *hash* para a *palavra* "som"=
$115 \times 256^2 + 111 \times 256^1 + 109 \times 256^0 = 7565165$

Metodologia

Algoritmo Trivial e Karp-Rabin

- Trivial: $O((|text| - |pattern|) \times |pattern|) = O(|text| \times |pattern|)$
- Karp-Rabin: $O((|text| - |pattern|)$

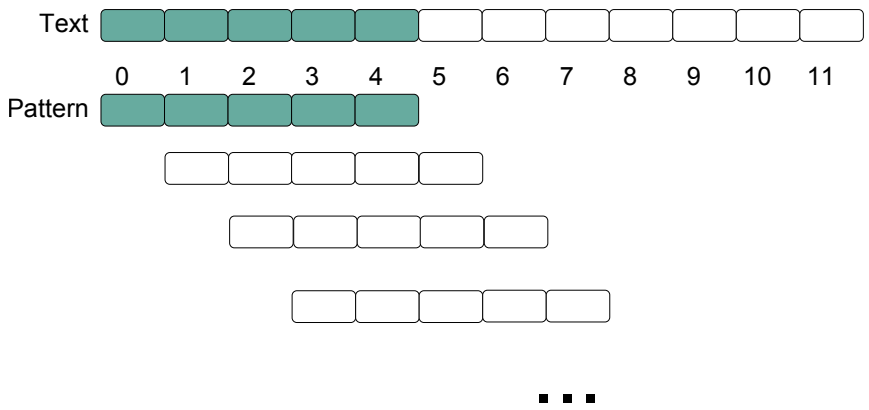


Figura: Buscando um padrão contendo cinco *letras* em um texto. *Subcadeias* do texto de cinco *letras* são comparadas ao padrão.

Metodologia
Outras definições

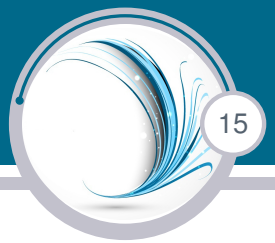
- Casamento
- Tile
- Letra marcada
- Casamento maximal
- Tamanho mínimo de casamento

Metodologia
Algoritmo *Greedy-String-Tilling*

- Busca *casamentos maximais* para a posterior criação de um *tile*.
- Guloso.

Metodologia

Algoritmo Running-Karp-Rabin Greedy-String-Tilling



- Busca *casamentos maximais* para a posterior criação de um *tile*.
- Rolling Hash.
- Previsão estática.

Abordagem
Objetivos

Objetivos do trabalho:

- Detecção de plágio em textos simples utilizando RKR-GST;
- Linguagem C++;
- Possibilitar fácil adaptação para detecção de plágio em códigos-fonte;
- Licença GNU GPL; e
- Programa bem documentado.

Decisões de projeto:

- Classes paramétricas;
- Base de tamanho 2;
- Uso do Doxygen; e
- Licença GNU GPL e programa bem documentado.

Abordagem
Medida de similaridade

$$similaridade(a, b) = (2 * numberOfTokensTiled)/(length(a) + length(b))$$

Abordagem
Testes realizados

Casos de teste divididos em seis grupos. Cada grupo com os seguintes arquivos:

- `resume.txt`;
- `original.txt`;
- `reordering.txt`;
- `redundancy.txt`; e
- `redundancyAndReordering.txt`.

Tamanho mínimo de casamento = valor 7.
Tamanho de *casamento* inicialmente buscado = 10.

Considerações finais
Trabalhos futuros

Extensão do Rorschach para detecção de plágio em códigos-fonte.

- Alterações na classe Reader:
  - Remoção de comentários
  - Tokenização
  - Método apropriado
- Criação da classe Token
  - Sobrecarga de operadores

O código-fonte, a sua documentação e os casos de teste utilizados foram disponibilizados[1] via GitHub.

[1] Endereço eletrônico: https://github.com/iruynarak/rorschach.

Perguntas?