

M.G.V. NUNES\*

## SUMARIO

O Método de Harrison [2] para a busca exata de subcadeias de caracteres em textos utiliza o conceito de assinaturas associadas ao texto e às subcadeias procuradas. Com isso, além de compactar sensivelmente o texto original, sob esse processo nem todo o texto precisa ser exaustivamente pesquisado, isto é, apenas aqueles trechos do texto original onde há uma alta probabilidade de ocorrência da subcadeia são realmente pesquisados caracter por caracter.

Este trabalho apresenta dois tipos de funções-assinatura amplamente testadas e os resultados experimentais obtidos.

## ABSTRACT

Harrison's Method [2] uses text signatures in String Searching. With this technique there is a sensible compactation of the original text and is not necessary to search the text wxhaustively. The text signature is examined first to determine if there is a high probability of finding the sought pattern and only in this case the correspondent line of text is scanned.

In this work we study and test in deep two types of text signatures and show the experimental results obtained.

\*Bacharel em Ciências da Computação (UFSCar-1980). Mestre em Ciências da Computação (USP-1985); algoritmos, sistemas especialistas; ICMSC-USP - Caixa Postal, 668 - São Carlos, SP - CEP 13560; Tel:(0162) 712214.

O problema de detetar a (não) ocorrência de uma ou várias cadeias de caracteres num texto caracteriza uma parte importante de muitos problemas relacionados a processamento de textos como edição de textos, recuperação de informações, manipulação de símbolos e pesquisa bibliográfica.

Vamos analisar, neste trabalho, um método de busca apropriado para textos grandes e pouco dinâmicos, uma vez que é realizado um pré-processamento do texto, codificando-o e compactando-o, e qualquer alteração no texto original requereria um novo pré-processamento. O objetivo principal do método é pesquisar exaustivamente uma parte do texto (por exemplo, uma linha) apenas quando houver uma grande probabilidade de se encontrar a cadeia neste trecho. Neste caso, qualquer método direto de busca pode ser utilizado (ressaltamos o método de Boyer e Moore [1]).

### O MÉTODO DE HARRISON

A informação contida em cada linha do texto é condensada num código de  $w$  bits que passa a ser a assinatura da linha. O código é obtido aplicando-se uma função hash  $h$  a todos os grupos consecutivos de  $k$  símbolos, produzindo um inteiro entre 0 e  $w-1$ . O bit correspondente ao inteiro produzido torna-se 1. Esta cadeia binária resultante, que contém 1 nas posições que correspondem a certas subcadeias de comprimento  $k$ , é chamada  $k$ -assinatura da linha do texto. O mesmo procedimento é efetuado para obter a  $k$ -assinatura da cadeia a ser procurada. Compara-se então, a assinatura da cadeia com as assinaturas das linhas e apenas quando este teste sugerir uma provável correspondência na linha associada, é que esta linha original será realmente pesquisada.

### ESCOLHA DOS PARÂMETROS $w$ E $k$ E DA FUNÇÃO HASH $h$

Para um texto sobre um alfabeto de 64 símbolos, tomamos  $w$ , o tamanho da assinatura, múltiplo de 8 e 64 e 128. Em nossos experimentos, tomamos  $k = 2$  e 3 para a busca de cadeias com mais de 1 e 2 caracteres, respectivamente. Lembramos que  $k > 1$  nos fornece informações sobre a posição do carácter na linha.

A função sugerida por Tharp e Tai [4] exige que os símbolos do texto sejam divididos em classes baseadas na frequência desses símbolos no texto, tal que as probabilidades de ocorrência são aproximadamente iguais para cada classe. Para um par de símbolos  $y_1y_2$ , a função hash é  $h(y_1y_2) = \text{número-de-classes} * T[y_1] + T[y_2]$ , onde o número-de-classes é escolhido como sendo o maior inteiro,  $m$ , tal que  $m^k \leq w$  e  $T[y_i]$  fornece o número da classe ( $0 \leq m-1$ ) a qual pertence o símbolo  $y_i$ .

Observamos que quanto maior o alfabeto maior deve ser  $w$ , para obtermos um maior número de classes de símbolos.

A segunda função hash testada é uma função simples que, ao contrário da anterior, não requer qualquer informação prévia sobre o texto. Ela simplesmente toma a somatória das or-

dens dos caracteres da subcadeia de comprimento  $k$  no conjunto de caracteres adotado pelo compilador Pascal, módulo o comprimento da assinatura.

## RESULTADOS EXPERIMENTAIS

Para grupos de 50 cadeias arbitrárias de comprimentos variando de 2 a 15 caracteres, obtivemos o número médio de linhas retidas no teste de assinaturas, e portanto pesquisadas exaustivamente, e dentre estas, o número médio de linhas em que a busca resultou em sucesso, ou seja, onde a cadeia foi encontrada.

De uma maneira geral, 3 fatores influenciaram no comportamento do método: o comprimento da assinatura,  $w$ , a função hash aplicada e o comprimento da cadeia procurada.

Quanto maior a assinatura, menor o número de linhas pesquisadas exaustivamente, no sentido de que menor número de cadeias são identificadas incorretamente como subcadeias pelo teste de assinaturas. Em consequência disso, temos um correspondente acréscimo no número de linhas com sucesso. Quanto maiores as cadeias procuradas, mais raramente elas ocorrem no texto e portanto, um menor número de linhas é retido no teste de assinaturas.

A função hash mais simples, que não exige informações sobre o texto, mostrou-se em média inferior à de Tharp e Tai quanto maior a assinatura adotada, pois neste caso, o número de classes de símbolos também é maior. Mas, no caso  $k = 3$ , para assinaturas menores (64 bits) a função mais simples comportou-se em média, de maneira superior. Isto se deve ao fato de que no caso  $k = 3$  o número de classes de símbolos para a função de Tharp e Tai diminuiu causando com isso um acréscimo na probabilidade de ocorrência de cada classe. Assim, quando dispusermos de espaço para assinaturas longas e de informações sobre frequências de símbolos no texto, optamos pela função de Tharp e Tai. Se, por outro lado, não tivermos informações sobre o texto ou necessitarmos usar assinaturas menores ou ainda se desejarmos procurar por cadeias com mais de 2 caracteres, devemos optar por uma função mais simples, como a descrita anteriormente.

Mais detalhes sobre o experimento bem como os gráficos que ilustram os resultados obtidos, encontram-se em [3-Cap.5].

## BIBLIOGRAFIA

- [1] Boyer, R.S., Moore, J.S. - "A Fast String Searching Algorithm" - Comm. of ACM - Vol. 20 - nº 10 - pp. 762-772 - Out.77.
- [2] Harrison, C.M. - "Implementation of the Substring Test by Hashing" - Comm. of ACM - vol. 14 - nº 12 - pp. 777-779 - Dez.71.
- [3] Nunes, M.G.V. - "Algoritmos de Busca em Texto" - Dissertação de Mestrado - ICMSC-USP, São Carlos, SP - Jan.85.
- [4] Tharp, A.L., Tai, K. - "The Practicality of Text Signatures for Accelerating String Searching" - Software-Practice and Experience - vol. 12 - pp. 35-44 - 1982.