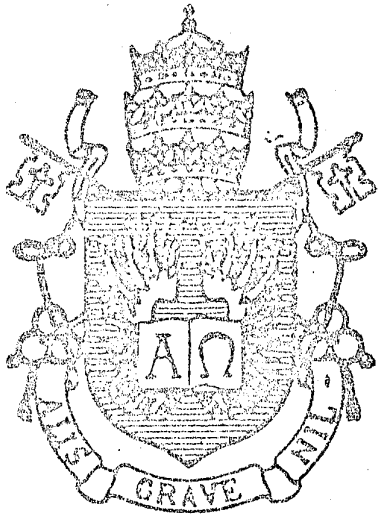

PUC

Série: Monografias em Ciência da Computação, 17/88

UM ALGORITMO DE HIFENIZAÇÃO MULTILÍNGUE

CLOVIS TORRES FERNANDES

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

A MARQUÊS DE SÃO VICENTE, 225 – CEP 22453

RIO DE JANEIRO – BRASIL

PUC/RJ - DEPARTAMENTO DE INFORMÁTICA

Série: Monografias em Ciência da Computação, 17/88

Editor: Paulo Augusto Silva Veloso outubro, 1988

UM ALGORITMO DE HIFENIZAÇÃO MULTILÍNGUE

CLOVIS TORRES FERNANDES*

* De licença da Divisão de Ciências de Computação do ITA - Instituto Tecnológico de Aeronáutica; trabalho parcialmente financiado por CAPES/PICD.

Responsável por publicações:

Rosane Teles Lins Castilho
Assessora de Biblioteca, Documentação e Informação
PUC/RJ - Depto. de Informática
Rua Marquês de São Vicente, 225 - Gávea
22453 - Rio de Janeiro, RJ
Brasil

RESUMO

Este trabalho apresenta um algoritmo de hifenização multilíngue automática. Com comandos do formatador, o usuário pode informar ao hifenizador a língua atual do texto. O algoritmo funciona bem para línguas com regras bem definidas, como o português, bem como para línguas com hifenização difícil, como o inglês. Mostram-se algumas idéias para aperfeiçoar o processo de hifenização. Também mostram-se como as estruturas dos idiomas foram armazenadas numa implementação real.

Palavras-chaves: formatador de textos, separação silábica, hifenização multilíngue, hifenização para o inglês.

ABSTRACT

This report presents an automatic hyphenation algorithm for several languages. The user can inform the hyphenator the current language of the text through formatting commands. The algorithm works well for languages with well-defined rules, such as Portuguese, as well as for languages with difficult hyphenation, such as English. A few ideas are shown for improving the process of hyphenation. It is also shown how the language structures are stored in a real implementation.

Keywords: text formatter, syllabic separation, multilingual hyphenation, English hyphenation.

SUMÁRIO

INTRODUÇÃO

O formatador de textos é um programa que, a partir da introdução de comandos entre blocos de texto, elabora a aparência física de um documento, de acordo com o formato desejado pelo usuário. Neste processo, o texto de saída é construído linha por linha. Quando a próxima palavra não se acomodar na linha de saída corrente e se deseje uma melhor qualidade do texto impresso, torna-se necessário fazer o ajuste da margem direita. Neste ponto temos, basicamente, duas soluções. A primeira consiste em intercalar ou suprimir espaços na linha corrente, de forma que a palavra anterior fique ajustada à direita, e passar a palavra sob consideração para o começo da linha seguinte. A segunda consiste em quebrar a palavra atual em duas partes, de tal forma que a primeira parte seja colocada ajustada à direita da linha atual e a segunda na próxima linha. Este processo chama-se hifenização.

Neste trabalho, o termo hifenização ainda significa que um vocábulo ou tem um ponto correto entre sílabas indicado ou tem todas as sílabas identificadas. Separação silábica significa que um vocábulo tem todas as sílabas identificadas. Caso tenha mais de uma sílaba determinada, o processo de hifenização determina o ponto de separação que melhor se ajuste na linha de saída corrente.

O processo de hifenização pode ser realizado ou não automaticamente. A maioria dos formatadores atuais permitem que o usuário determine o ambiente em que vai trabalhar, através de um comando

simples que habilita ou desabilita a hifenização. Neste artigo apresenta-se um algoritmo que realiza hifenização de vocábulos, automaticamente, para várias línguas. Neste caso, além de habilitar o processo de hifenização, é necessário indicar para o hifenizador a língua em que o texto está sendo escrito.

Nas seções seguintes mostram-se o processo de separação silábica para línguas como o português, francês, espanhol, italiano etc. e o processo de hifenização multilíngue que continua fazendo separação silábica das línguas citadas e ainda faz hifenização para línguas difíceis de hifenizar, cujo exemplo maior é o inglês.

O PROCESSO DE SEPARAÇÃO SILÁBICA MULTILÍNGUE

Em Fernandes (1995) apresentou-se um algoritmo eficiente que realiza separação silábica para qualquer língua que, como o português, possua regras de separação silábica bem definidas e com poucas exceções. O algoritmo básico baseia-se num transdutor a pilha e é dirigido por uma tabela de transição de estados. A tabela apresenta de forma codificada as regras de separação silábica de um idioma em particular, aqui denominada tabela de relacionamentos. Dado que o algoritmo é dirigido pela tabela de relacionamentos, para se realizar a separação silábica para uma língua, basta carregar a tabela codificada da mesma.

O transdutor compõe-se de 4 estados (inicial, sílaba não formada, sílaba formada ou sujeita a alteração, final) e cincos ações que realizam a transição de estados, as quais posicionam os hífens nos locais corretos, segundo as regras estabelecidas na

composição de sílabas da língua em questão. Dada a letra anterior (la) e a letra correntemente em exame (le), cada ação (relacionamento) pode ser descrita da seguinte maneira:

- ação 0 - junta le à cadeia de saída;
- ação 1 - separa le da sílaba anterior;
- ação 2 - inicia nova sílaba com le, adicionando la à sílaba anterior;
- ação 3 - junta le à última sílaba formada quando consoante;
- ação 4 - junta numa mesma sílaba quando no início da palavra, caso contrário idêntico à ação 2.

A figura 1 apresenta a tabela de relacionamentos para a língua francesa para se ter uma idéia da composição das tabelas. A linha corresponde a elementos de la e a coluna a de le. Com estas duas entradas obtém-se o número da ação a ser tomada.

Encontros consonantais são tratados como uma única consoante, com ação 0 . Em francês, por exemplo, se la = 'G' e le = 'N' a ação será a 0 e teremos 'GN' numa única sílaba. Ainda em português, encontros consonantais tais como 'PS', 'PN' etc. são tratados com ação 4, ou seja, quando no início da palavra, permanecem na mesma sílaba (por exemplo em psi-co-se), caso contrário a primeira consoante fica numa sílaba e a segunda numa outra (por exemplo em cáp-su-la).

TABELA['G', 'N'] = 0

LE ↓

LA ----►

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	1	1	1	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	0	1
B	0	2	2	2	0	2	2	2	0	2	2	0	2	2	0	2	2	0	2	2	0	2	2	2	0	2
C	0	2	2	2	0	2	2	2	0	2	2	0	2	4	0	2	2	0	2	2	0	2	2	2	0	2
D	0	2	2	2	0	2	2	2	0	4	2	4	2	2	0	2	2	0	2	2	0	2	2	2	0	2
E	0	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	0	1
F	0	2	2	2	0	2	2	2	0	2	2	0	2	2	0	2	2	0	2	2	0	2	2	2	0	2
G	0	2	2	2	0	2	2	0	0	2	2	0	2	0	0	2	2	0	2	2	0	2	2	2	0	2
H	0	2	2	2	0	2	2	2	0	2	2	2	2	2	0	2	2	0	2	2	0	2	2	2	0	2
I	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	0	1
J	0	2	2	2	0	2	2	2	0	2	2	2	2	2	0	2	2	2	2	2	0	2	2	2	0	2
K	0	2	2	2	0	2	2	0	0	2	2	0	2	2	0	2	2	2	2	2	0	2	2	2	0	2
L	0	2	2	2	0	2	2	2	0	2	2	2	2	2	0	2	2	2	2	2	0	2	2	2	0	2
M	0	2	2	2	0	2	2	2	0	2	2	2	2	4	0	2	2	2	2	2	0	2	2	2	0	2
N	0	2	2	2	0	2	2	2	0	2	2	2	2	2	0	2	2	2	2	2	0	2	2	2	0	2
O	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1
P	0	2	2	2	0	2	2	0	0	2	2	0	2	4	0	2	2	0	4	4	0	2	2	2	0	2
Q	0	2	2	2	0	2	2	2	0	2	2	2	2	2	0	2	2	2	2	2	0	2	2	2	0	2
R	0	2	2	2	0	2	2	0	0	2	2	2	2	2	0	2	2	2	2	2	0	2	2	2	0	2
S	0	2	4	2	0	2	2	0	0	2	0	4	4	4	0	4	4	2	2	4	0	4	2	2	0	2
T	0	2	2	2	0	2	2	0	0	2	2	2	4	2	0	2	2	0	4	2	0	2	2	2	0	4
U	0	1	1	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1
V	0	2	2	2	0	2	2	2	0	2	2	2	2	2	0	2	2	0	2	2	0	2	2	2	0	2
W	0	2	2	2	0	2	2	0	0	2	2	2	2	2	0	2	2	0	2	2	0	2	2	2	0	2
X	0	2	2	2	0	2	2	2	0	2	2	2	2	2	0	2	2	2	2	2	0	2	2	2	0	2
Y	0	1	1	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1
Z	0	2	2	2	0	2	2	2	0	2	2	2	2	2	0	2	2	2	2	2	0	2	0	2	0	2

FIG. 1 - TABELA DE RELACIONAMENTOS PARA O FRANCÊS.

Como exemplo do funcionamento do algoritmo, mostra-se a separação silábica do vocábulo francês 'grognement'. Deve-se notar que o algoritmo trabalha com dois caracteres brancos virtuais agregados a todos vocábulos de entrada, um no início e o outro no fim . O vocábulo de entrada é visto assim: 'Bgrognement B'. A tabela de relacionamentos usada é a do francês. Inicialmente temos la = 'B' (branco) e le = 'g' que dá, por definição, a ação 0 e 'g' é colocado na saída; em seguida, temos la = 'g' e le = 'r' que, dá a ação 0 pois TABELA['G','R'] = 0 e 'r' é colocado ao lado

de 'g' na saída; quando la = 'o' e le = 'g', pela tabela temos ação 1 e '-g' é colocado ao lado de 'o' na saída; quando la = 'n' e le = 't', temos ação 2, o que significa que 'n' deve ser incorporado à sílaba anterior e 't' fazer parte de nova sílaba; e, finalmente, quando la = 't' e le = 'Ø' temos ação 3 que coloca 't' na sílaba anterior, pois todas as consoantes que encerram um vocábulo fazem parte da sua última sílaba. O desenvolvimento completo da separação silábica é o seguinte, onde a numeração indica as ações e sub-ações tomadas:

'la'	'le'	ação	hifenização
Ø	g	### ação 0 ###	g
g	r	### ação 0 ###	gr
r	o	### ação 0 ###	gro
o	g	### ação 1 ###	gro-g
g	n	### ação 0 ###	gro-gn
n	e	### ação 0 ###	gro-gne
e	m	### ação 1 ###	gro-gne-m
m	e	### ação 0 ###	gro-gne-me
e	n	### ação 1 ###	gro-gne-me-n
n	t	### ação 2 ###	
		2.1	gro-gne-me-n
		2.2	gro-gne-me
		2.3	gro-gne-men-t
t	Ø	### ação 3 ###	
		3.1	gro-gne-men-t
		3.2	gro-gne-men
		3.3	gro-gne-ment

O acesso à tabela de relacionamentos é controlado por uma função, aqui denominada AÇÃO, que determina a próxima ação que dirige o algoritmo de separação silábica. Muitas vezes a função AÇÃO determina a próxima ação sem acessar a tabela de relacionamentos. Tal se dá, por exemplo, no início da separação, quando a primeira letra da palavra a ser separada deve ser introduzida na cadeia de caracteres que representa a palavra já hifenizada.

É na função AÇÃO que se realiza o tratamento de hiatos. Dado que se trabalha com vogais acentuadas, pode-se determinar o relacionamento entre elas para se obter a ação apropriada.

A tabela de relacionamentos é subscrita pelos caracteres maiúsculos sem acento, assim a função AÇÃO funciona como um filtro, colocando os caracteres de entrada (quer sejam letras acentuadas, maiúsculas ou minúsculas, ç ou Ç) na faixa correta: letras maiúsculas sem acento.

Como se pode notar, a função AÇÃO esconde detalhes específicos de cada língua. Assim, para cada língua, deve-se projetar uma função AÇÃO correspondente.

A separação silábica multilíngue opera da seguinte forma. A nível do formatador deve-se possuir um comando que especifica em que língua o texto está sendo escrito. O que tal comando faz é carregar a tabela de relacionamentos e especificar a função AÇÃO correspondentes. A partir deste instante, a separação silábica será realizada de acordo com as regras embutidas na tabela e na função. Ao se desejar trocar de língua, deve-se

especificar o comando para o novo idioma.

Deve-se ressaltar que, por mais elaboradas que sejam as tabelas de relacionamentos e as funções AÇÃO, ainda assim aparecem vocábulos, em todas as línguas em que o algoritmo se aplica, que não são separados corretamente. Tais palavras constituem exceções , ou seja, não estão sujeitas às regras de separação silábica da língua em questão. No português, o algoritmo não consegue separar corretamente palavras com hiatos não acentuados (como ca-ir), encontros vocálicos (como po-ei-ra) e palavras verdadeiramente exceções (como sub-lo-car). Para efeito deste sistema de separação silábica, constitue exceção todo vocábulo que o algoritmo não consegue fazer a separação silábica corretamente.

Pode-se dar tratamento às exceções basicamente de duas maneiras. A nível de formatador utilizando-se caracteres que indicam possíveis pontos de separação ou a nível de separador. Este trabalho propõe-se a explorar o segundo tratamento mas tendo em vista que ambos podem ser compatibilizados em um formatador real.
A seção seguinte mostra como estender este algoritmo para se poder realizar hifenização do inglês e de outras línguas que não apresentem regras de separação silábica bem definidas.

HIFENIZAÇÃO PARA O INGLÊS

O inglês será usado como exemplo de língua que não possua regras sistemáticas de separação silábica. Este é um idioma muito difícil para se hifenizar corretamente, desde que quase

toda regra geral tem exceções. Por exemplo, 't' e 'h', nesta ordem, sempre pertencem à mesma sílaba como regra geral, como em 'mo-ther', mas tem-se exceção como 'light-house'. Além disso há problemas semânticos que não podem ser resolvidos apenas por inspecionar os padrões de letras. Um bom exemplo é 'present' verbo que é dividido 'pre-sent' e 'present' substantivo que é dividido 'pres-ent'.

Apesar da versatilidade e eficiência do algoritmo anterior de separação silábica, em línguas onde o esquema de separação seja mais complexo, como no inglês, não se obterá o mesmo desempenho utilizando esse algoritmo. Torna-se necessário projetar um modelo que envolva tanto estas linguagens, quanto aquelas que já foram satisfeitas pelo algoritmo acima.

Há duas técnicas principais que têm sido usadas para hifenização do inglês em sistemas automatizados. A primeira consiste em fornecer um hifenizador lógico que é um algoritmo que pesquisa uma palavra para pontos de hifenização. O problema com esta abordagem é que os algoritmos viáveis obtêm uma margem de hifenização correta entre 60 e 70%. O segundo é manter um dicionário de palavras junto com seus possíveis pontos de separação. A desvantagem desta técnica é a necessidade de grande quantidade de memória para armazenar o dicionário e o tempo de procura dos pontos de hifenização para os vocábulos.

Na prática, um método híbrido é frequentemente empregado, utilizando tanto um algoritmo quanto um dicionário de exceções. O dicionário de exceções deve cobrir os 30 a 40% que o algo-

ritmo não hifeniza corretamente. Mas mesmo neste caso se necessitaria de grande quantidade de memória para armazenar o dicionário de exceções.

Um modo de aliviar este problema é buscar maneiras práticas de hifenização que operem com um dicionário de exceções pequeno (+/- 300 palavras) e apresentem resultados semelhantes aos obtidos com um dicionário de exceções bem maior. Uma prática muito difundida (Moitra et al., 1979; Knuth, 1979) utiliza listas de sufixos e prefixos usuais como padrões que indicam um único ponto de hifenização: após um prefixo ou antes de um sufixo. Se se acrescentar palavras raízes de palavras compostas a estas listas (ex. 'pothole ---> pot-hole', 'lighthouse ---> light-house' e 'fireside ---> fire-side'), diminui muito a necessidade de manter um dicionário de exceções grande. De fato, o dicionário de exceções só deve manter palavras que não sejam hifenizadas corretamente com o algoritmo separador e com a pesquisa das listas. As listas de afixos mostraram-se ser de grande utilidade para se atingir maior precisão e velocidade no processo de hifenização para o inglês.

Assim, um modelo que viabiliza o processo de hifenização para o inglês deve constar de um dicionário de exceções, de listas de sufixos e prefixos e de um algoritmo lógico de separação silábica. Para que este modelo inclua o algoritmo anterior é preciso que haja compatibilidade entre os algoritmos de separação silábica.

Uma solução que é compatível com esse modelo é a encontrada na rotina de hifenização para o inglês apresentada por Moitra et al. (1979). Basicamente ele fundamenta o seu algoritmo na procura de certos padrões de cadeias de caracteres ao longo da palavra em exame. Falhando esta pesquisa, ele hifeniza a palavra através de uma "tabela de quebra" (figura 2), similar à tabela de relacionamentos. Esta tabela codifica as duas ações que podem ser tomadas, estatisticamente, em relação à letra em exame: juntá-la à letra anterior (ação 0) ou separá-la da letra anterior através de hífen (ação 1). Estas ações correspon-

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
B	0	1	1	1	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1
C	0	1	1	1	0	1	1	0	0	1	0	0	1	1	0	1	1	0	1	0	0	1	1	1	1	0
D	0	1	1	1	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
F	0	1	1	1	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	1	1	1	0	1
G	0	1	1	1	0	1	1	0	0	1	1	0	1	0	0	1	1	0	1	1	0	1	1	1	1	1
H	0	1	1	1	0	1	1	1	0	1	1	1	1	0	1	1	1	0	1	0	0	1	1	1	1	1
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
J	0	1	1	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1
K	0	1	1	1	0	1	1	1	0	1	1	1	1	0	0	1	1	1	0	1	0	1	1	1	1	1
L	0	1	1	0	0	1	1	1	0	1	0	0	1	1	0	1	1	1	0	1	0	0	1	1	0	1
M	0	1	1	1	0	1	1	1	0	1	1	1	1	1	0	0	1	1	1	1	0	1	1	1	1	1
N	0	1	0	0	0	1	0	1	0	1	0	1	1	1	0	1	1	1	0	0	0	1	1	1	1	1
O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
P	0	1	1	1	0	1	1	0	0	1	1	0	1	1	0	1	1	0	1	0	0	1	1	1	1	1
Q	0	1	1	1	0	1	1	1	0	1	1	1	1	1	0	1	1	0	1	1	0	1	1	1	0	1
R	0	1	0	0	0	1	0	1	0	1	1	1	0	0	0	1	1	1	0	0	0	0	1	1	0	0
S	0	1	0	1	0	1	1	0	0	1	0	1	1	0	0	0	0	1	0	0	0	1	0	1	1	1
T	0	1	0	1	0	1	1	0	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	1
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
V	0	1	1	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1
W	0	1	1	1	0	1	1	0	0	1	1	1	1	1	0	1	1	0	0	1	0	1	1	1	0	1
X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1
Y	0	1	1	1	1	1	1	1	0	1	1	0	0	0	0	1	1	0	0	0	0	1	1	1	0	0
Z	0	1	1	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1

FIG. 2 - TABELA DE RELACIONAMENTOS PARA O INGLÊS.

dem exatamente às ações 0 e 1 do algoritmo anteriormente citado, ou seja, pode-se usá-lo com esta tabela sem qualquer modificação.

Isto torna possível agregar este algoritmo ao anterior, tornando-o mais amplo e geral, aplicável a línguas cuja separação silábica seja ditada por regras, como o português e àquelas que não possuam regras sistemáticas como o inglês. O algoritmo assim ampliado pode, então, ser aplicado às mais diversas línguas: português, inglês, espanhol, francês, italiano etc. Deste ponto em diante este algoritmo será referenciado como algoritmo de hifenização ou hifenizador, possuindo a seguinte estrutura:

início

- se o usuário deseja, então pesquisa dicionário de exceções;
- se anterior falhou e usuário deseja, então pesquisa se vocábulo possui sufixo da tabela de sufixos;
- se anterior falhou e usuário deseja, então pesquisa se o vocábulo possui prefixo da tabela de prefixos;
- se anterior falhou, então, usando a tabela de relacionamentos, aplica o algoritmo de separação silábica;
- procura melhor posição de hifenização.

fim

Inicialmente, a pesquisa no dicionário de exceções é necessária para não se perder tempo processando uma palavra reconhecidamente exceção. É uma estrutura que deve ser ampliada à medida que se usa a rotina de hifenização. A pesquisa no dicionário pode apresentar dois resultados:

- fracasso - o vocábulo em exame não consta do dicionário;
- sucesso - o vocábulo existe e obtém-se os pontos de hifenização registrados para o mesmo; se, por exemplo, o vocábulo em questão for 'inelegant', obtém-se 'in-el-e-gant'.

Caso esta fracasse, verifica se o vocábulo possui afixos que constem das listas de sufixos e prefixos (vide resumo da lista de afixos na figura 3). A pesquisa com a lista de sufixos também pode apresentar dois resultados:

- fracasso - o vocábulo não possui sufixos que constam da lista;
- sucesso - o vocábulo possui sufixo da lista e obtém-se o ponto de hifenização; se, por exemplo, o vocábulo for 'stubbornness', obtém-se que 'ness' é sufixo e a hifenização é 'stubborn-ness'.

Acontece de forma semelhante para a pesquisa com a lista de prefixos. Por exemplo, se o vocábulo for 'overcome', obtém-se que 'over' é um prefixo e a hifenização é 'over-come'.

SUFIXOS		PREFIXOS	
CY	ABLE	MICRO	OVER
DOM	ABLY	AB	DI
DAY	ANCE	ABS	SHIP
FIC	BOOK	ACTU	SHOP
FY	CADE	AD	DIN
GY	CANT	WORK	SIDE
STAND	CASE	AG	DOC
LY	CATE	MONO	EARTH
ADE	HALF	AN	ECON
AGE	HOLD	ANTE	ELEC
LINE	HOLE	ANTI	EMB
TIC	SHIP	ANY	UNDER
TER	SHOP	AP	EPI
PLA	TION	ARBI	ES
PLE	TISE	AU	EVERY
PLY	TIVE	AUTO	EX

FIG.3 - RESUMO DAS LISTAS DE AFIXOS.

Caso estas fracassem, usa-se a tabela de relacionamentos para o inglês, aplicando-se o algoritmo de separação silábica de Fernandes (1983), com a função AÇÃO apropriada. O resultado será um vocábulo com todos os pontos de separação silábica indicados, de acordo com as regras embutidas na tabela de relacionamentos. Por exemplo, se o vocábulo for 'bowlegged', obtém-se todos pontos de separação silábica: 'bow-leg-ged'. Utilizando a tabela de relacionamentos para o inglês, tem-se a seguinte se-

quência de separação silábica para o vocábulo 'bowlegged':

'la'	'le'	ação	hifenização
ƀ	b	### ação 0	### b
b	o	### ação 0	### bo
o	w	### ação 0	### bow
w	l	### ação 1	### bow-l
l	e	### ação 0	### bow-le
e	g	### ação 0	### bow-leg
g	g	### ação 1	### bow-leg-g
g	e	### ação 0	### bow-leg-ge
e	d	### ação 0	### bow-leg-ged
d	ƀ	### ação 0	### bow-leg-ged

Pode-se notar certa semelhança entre a tabela de relacionamentos e a tabela de quebra; entretanto, a diferença é notória, já que a elaboração da tabela de relacionamentos exige uma sistemática baseada nas regras de separação silábica, enquanto a tabela de quebra dá a probabilidade estatística de hifenização entre toda possivel combinação de duas letras. Quanto ao uso, no entanto, ambas são vistas de forma idêntica pelo algoritmo separador e tratadas como indicadores de ações a serem tomadas.

A nível de formatador, o usuário pode indicar se deseja ou não que as pesquisas sejam realizadas nos dicionários de exceções e/ou nas listas de afixos. Em português, por exemplo, pode-se dispensar o uso do dicionário de exceções e de listas de afixos.

Cada língua pode possuir uma estrutura com as seguintes sub-estruturas: dicionário de exceções, lista de sufixos, lista de prefixos, tabela de relacionamentos (figura 4). Tem-se uma estrutura geral (figura 5) onde a flexibilidade de aplicação é fator primordial para sua implementação e seu uso real. Com um comando do formatador indica-se a língua corrente em que o texto está sendo escrito. Com a execução deste comando, carrega-se a estrutura correspondente à mesma, não somente a tabela de relacionamentos (com a função AÇÃO apropriada indicada), mas também, se existirem e caso o usuário deseje, o dicionário de exce-

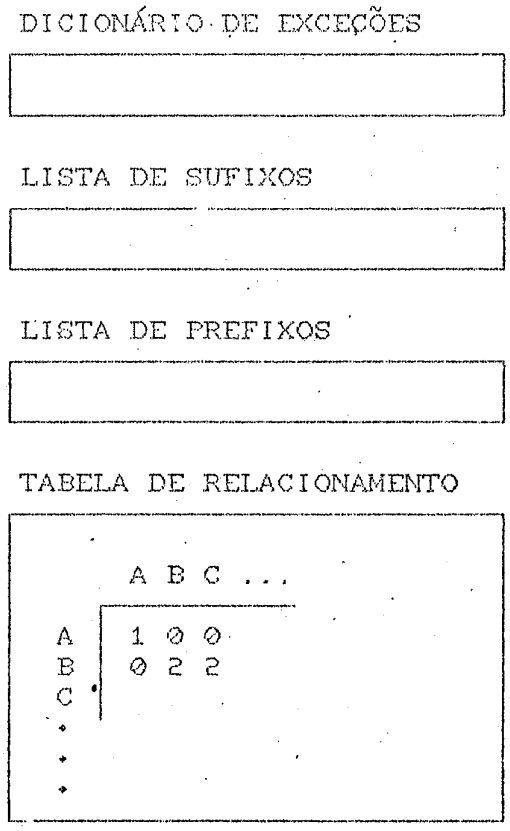

FIG. 4 - ESTRUTURA PARA HIFENIZAÇÃO NUMA LÍNGUA.

ções e as duas listas, que fornecem uma grande possibilidade de sucesso na tentativa de hifenização, através do algoritmo de hifenização único.

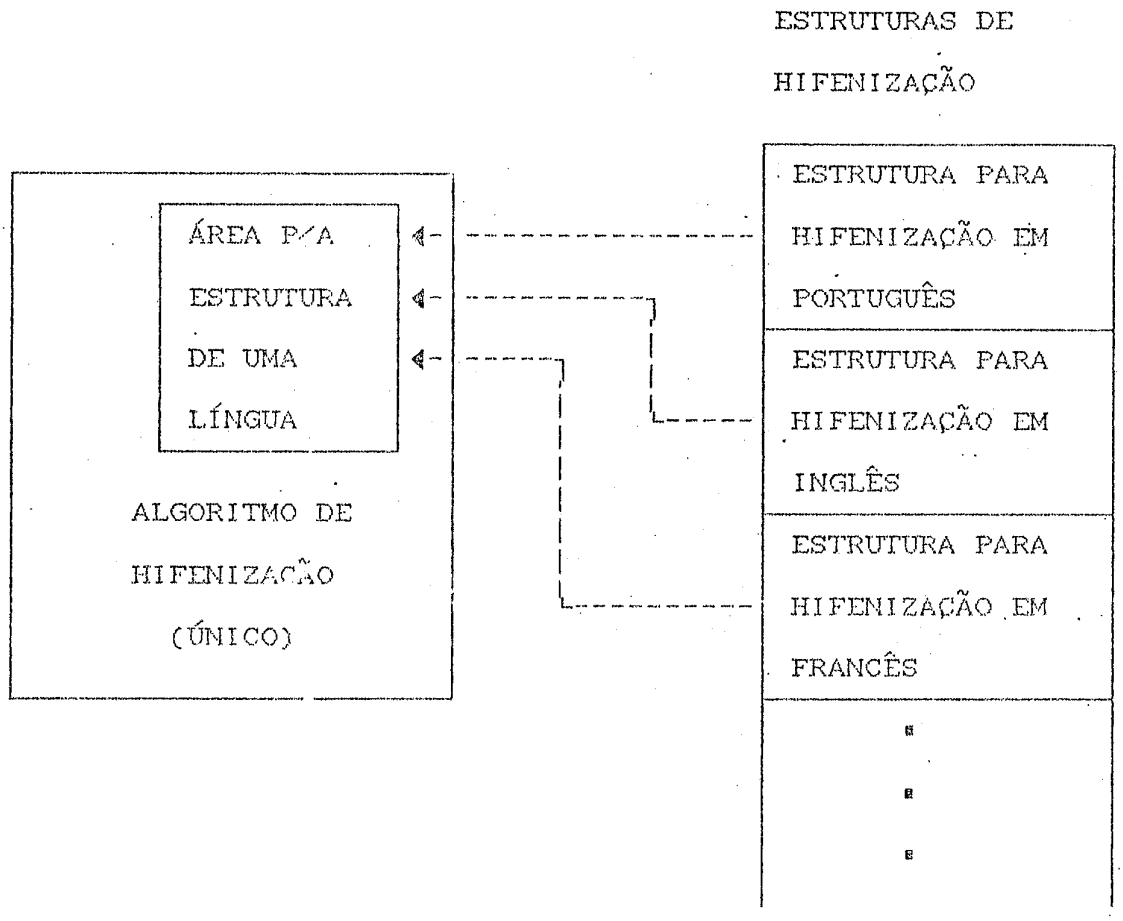

FIG. 5 - ESTRUTURA GERAL DO HIFENIZADOR.

APERFEIÇOANDO O PROCESSO DE HIFENIZAÇÃO

Pode-se aperfeiçoar o processo de hifenização em três níveis:

- ao nível do hifenizador, de forma a reduzir o tempo de processamento e obter-se menor número de palavras não hifenizadas corretamente.

- ao nível da geração automatizada da rotina de hifenização e
- ao nível do formatador, no que diz respeito à utilização mais eficaz do hifenizador.

A seguir apresentam-se algumas idéias encontradas em outros sistemas de hifenização que sugerem aperfeiçoamentos em um ou mais dos níveis acima.

Knuth (1979) apresenta um hifenizador para o inglês com uma estrutura muito parecida com a deste trabalho. A diferença é que ele procura encontrar todos os pontos de separação enquanto o hifenizador aqui apresentado fica satisfeito em encontrar um ponto ao menos. Knuth começa com um dicionário de exceções com cerca de 300 palavras. Se o vocábulo não se encontra no dicionário, ele verifica se ele possui algum sufixo de uma lista de sufixos; em seguida verifica se possui algum prefixo de uma lista de prefixos. Neste ponto a palavra se encontra particionada em três partes: prefixo, raiz e sufixo. A seguir, utilizando regras ad-hoc, hifeniza a raiz. Uma vez que a palavra toda esteja hifenizada, o algoritmo volta e acerta erros óbvios.

Este sistema utiliza rotinas para casamento de padrões para listas de sufixos e prefixos, que podem ser implementadas eficientemente usando máquinas de estado dirigidas por tabela. Como ele utiliza tabelas de afixos pequenas, as máquinas de estados também são pequenas. Apesar do sistema alvo deste trabalho utilizar tabelas de afixos maiores, pode-se pensar em utilizar essas rotinas para casamento de padrões com o objetivo de tornar

mais veloz a pesquisa nas tabelas de afixos.

Para o inglês, pode-se, ainda, utilizar neste trabalho regras ad-hoc para aperfeiçoar a função AÇÃO e para verificar se uma palavra já hifenizada contém determinados erros.

Machado (1986) apresenta um processo de separação silábica para o português que usa grafos direcionados para realizar a hifenização. Este esquema poderia ser adaptado ao algoritmo de hifenização objeto deste relatório, com o objetivo de se produzir e de se aperfeiçoar uma função AÇÃO para cada língua. Utilizando-se o seu processo, pode-se até pensar em se ter funções AÇÃO que constariam de um algoritmo único dirigido por tabela. As tabelas seriam obtidas automaticamente, uma para cada idioma.

Pringle (1981), no contexto de fotocomposição de textos por computador, apresenta um esquema de formatação de parágrafos que faz a justificação à direita com número mínimo de hífens, com o objetivo de aumentar a qualidade do texto impresso. Knuth (1979) também apresenta um esquema de formatação de parágrafos, mas com um algoritmo não tão simples como o de Pringle. O processo de Pringle consiste em colocar palavras para trás ou para frente, do ponto onde elas se encontram, dentro do parágrafo, com o uso de um algoritmo recursivo. Este comprime ou expande micro-espaços até limites razoáveis dentro de cada linha.

Com tal esquema pode-se pensar em sofisticar e aumentar a precisão do hifenizador com novas técnicas, visto que a necessidade

de hifenização torna-se reduzida e o tempo assim economizado pode ser utilizado no processo de hifenização. Assim, no mesmo tempo de processamento caso não utilizemos esse esquema, a chance de falha do hifenizador pode ser reduzida ou porque se utiliza menos o processo de hifenização ou porque, quando se o utiliza, o algoritmo pode ser mais sofisticado e dar respostas mais precisas na maioria das vezes.

A argumentação de Moitra et al. (1979) é que a hifenização não deve ser invocada a menos que absolutamente necessária, visto que o processo é computacionalmente muito caro. Esta também é a preocupação do esquema anterior mas Moitra et al. resolvem de maneira diferente, embora ambas possam e devam ser compatibilizadas. A nível de formatador deve existir um filtro com determinadas regras que procurem agilizar a utilização do hifenizador. Essas regras seriam formuladas com o objetivo de rejeitar, a priori, tantas palavras quanto possível e o mais cedo possível. Se não fôsse assim, algumas palavras passariam pelo hifenizador e sairiam sem serem hifenizadas, simplesmente porque seria impossível hifenizá-las. Como exemplo temos a palavra 'god'. Assim, por exemplo, palavras com apenas uma vogal ou palavras com três letras seriam rejeitadas. Cada língua ainda pode apresentar regras mais particulares. Por exemplo, em inglês, todas palavras com 4 letras não terminando em 'y' seriam rejeitadas.

Ainda está por se fazer uma pesquisa mais ampla, para cada língua alvo, para se determinar outras regras como essas acima,

que poderiam determinar com maior exatidão se uma palavra deve ou não passar pelo hifenizador.

ARMAZENAMENTO DAS ESTRUTURAS DE UMA LÍNGUA

Inicialmente, pensou-se num único arquivo contendo toda a estrutura para cada idioma, a fim de que, quando esta fosse indicada, houvesse o carregamento imediato do respectivo arquivo. Entretanto, como a rotina manipula um dicionário, supõe-se que este tenha expansão e manutenção periódicas. Logo deve-se usar um tipo de arquivo que possibilite este manuseio sem maiores esforços computacionais. Assim,os dicionários de exceções para todas a línguas foram colocados num único arquivo e as listas de afixos e a tabela de relacionamentos em arquivos diferentes, um para cada língua.

A estrutura do arquivo dos dicionários de exceções é a apresentada na figura 6. É uma estrutura encadeada com duas partes. Uma parte consiste numa estrutura de nós-cabeças que dão acesso aos diversos dicionários. Cada nó-cabeça identifica a língua e qual é o primeiro nó da lista de exceções. A outra parte consiste num conjunto de nós utilizados pelos diverosos dicionários, onde cada nó contém uma palavra exceção, as posições onde imediatamente antes devem ser inseridos hífens e um ponteiro para a localização do próximo nó.

Com esta estrutura pode-se acrescentar uma nova língua ao dicionário de exceções, criando-se um novo nó-cabeça. Pode-se, também, inserir uma nova palavra de qualquer dicionário no fim do arquivo ou em local apropriado se houve eliminação de

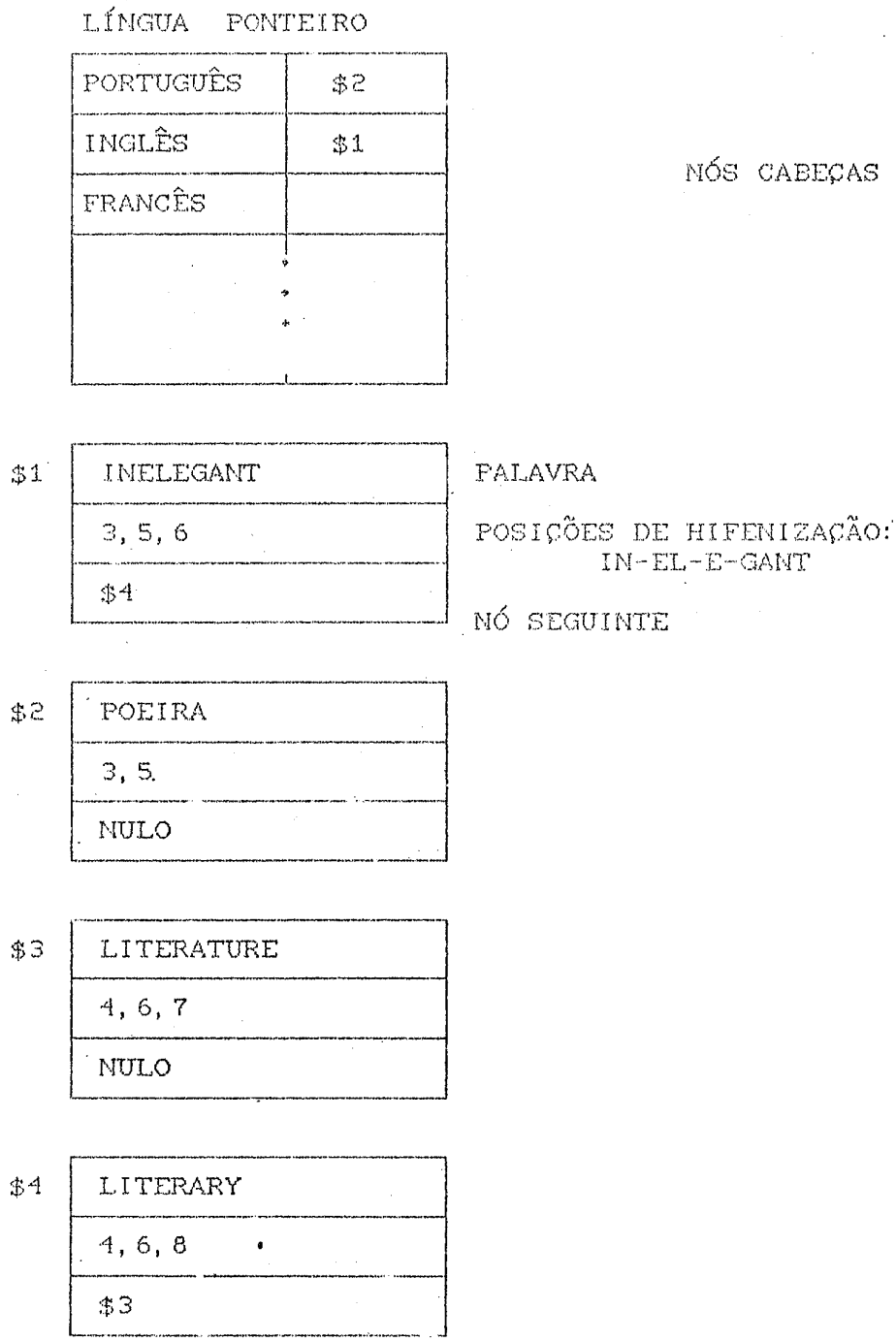

FIG. 6 - ESTRUTURA DO ARQUIVO DOS DICIONÁRIOS DE EXCEÇÕES.

palavras, ajustando-se depois os ponteiros para a localização ordenada da nova palavra. Logo, não é necesssário reescrever ou reordenar os dicionários. Deve-se ressaltar que esta estrutura não tem nada a ver com a estrutura de um dicionário específico na memória principal, em que deve permitir acesso direto mais rápido através de busca binária ou 'hashing', por exemplo.

O dicionário de exceções pode crescer ou diminuir de tamanho, de acordo com a vontade do usuário. Para tanto, fornece-se ao usuário uma rotina Mantém-Dicionário, que é externa ao sistema formatador e que lê num arquivo texto as mudanças necessárias.

A seguir mostra-se um arquivo exemplo de criação do dicionário de exceções:

```
.cria português p
.cria inglês i
.i
in-el-e-gant
.p
po-ei-ra
ba-i-nha
-bainha
.i
lit-er-a-ture
lit-er-ar-y
```

Na primeira linha cria-se o nó-cabeça para o português e 'p' indica que o texto que segue '.p' deve ser considerado como contendo exceções da língua portuguesa. O texto é considerado encerrado ao se encontrar um outro '.' na primeira coluna ou encontrar fim de arquivo. Idem para o inglês na segunda linha. Quando se deseja retirar uma palavra do dicionário basta prefixá-la com um hífen. A inserção ou eliminação de palavras no dicionário mantém a lista alfabeticamente ordenada. É interessante notar que a rotina Mantém-Dicionário poderia estar embutida num editor/formatador, visto que o trabalho que ela realiza nada mais é do que edição/formatação de texto.

CONCLUSÃO

Neste trabalho mostrou-se um sistema de hifenização multilíngue que opera adequadamente para línguas com regras bem definidas como o português, francês etc., bem como para línguas com hifenização difícil como o inglês. A principal característica do sistema é apresentar um algoritmo único que é dirigido por tabelas, as quais codificam as regras de hifenização para um determinado idioma.

Algumas idéias encontradas em sistemas de hifenização da literatura disponível (Knuth, 1979; Machado, 1986; Pringle, 1981; Moitra et al., 1979) sugerem aperfeiçoamentos ao trabalho apresentado. A maioria das idéias aqui apresentadas foram implementadas num formatador de textos experimental (Mioto, 1985) e revelaram-se muito eficientes. Espera-se que, com a difusão deste processo de hifenização, este seja incorporado a formatadores profissionais e

aperfeiçoado com a realimentação obtida com o seu uso prático bem como com novos avanços teóricos.

Reconhecimento: este relatório é fruto da colaboração de Tsutomu K. Murakami, Asiel Bonfim Filho e Paulo R. Mioto na implementação e dos colegas Edward Hermann Haeusler, Maria das Graças V. Nunes e Jeferson Simões Santana que leram a versão rascunho e deram sugestões muito úteis; no entanto, a responsabilidade das opiniões aqui emitidas é inteiramente do autor.

BIBLIOGRAFIA

FERNANDES, C.T. Um algoritmo de se-pa-ra-ção si-lá-bi-ca. 3º Simpósio sobre desenvolvimento de software básico. Rio de Janeiro, dez. 1983.

KNUTH, D.E. TEX and Metafont. New directions in Typesetting. Digital Press, 1979, pp.180-186.

MACHADO, R.J. Automação da divisão silábica em língua portuguesa. 3º Simpósio de Inteligência Artificial. Rio de Janeiro, 1986.

MIOTO, P.R. Formatador de textos. Trabalho de Graduação do ITA, São José dos Campos, dez. 1985.

MOITRA, A.; MUDUR, S.P.; NARWEKAR, A.W. Design and analysis of a hyphenation procedure. Software-Practice and Experience,9:325 - 337, Sep. 1979.

PRINGLE, A.L. Justification with fewer hyphens. The Computer Journal,24:4, 1981.