universidade de aveiro

Computação em Larga Escala

General Problems – Algorithmic analysis

António Rui Borges

Summary

- *Text processing in Portuguese*
  - *Characters encodings*
  - *Rules for text processing*
- *Determinant of a square matrix*
  - *Definition*
  - *Theorem of Laplace*
  - *Method of gaussian elimination*

Text processing in Portuguese - 1

Character encoding is essential to store and process textual information people routinely use to communicate among themselves. Many such codes were introduced in the computer world to express written contents as time went by.

The most popular ones are

- ASCII (*American Standard Code for Information Interchange*)

  it is a 7 bit code able to represent 95 graphical symbols and 33 control signals, which was used for many years to encode english language texts

- ISO/IEC 8859 (first published in 1987)

  it is a 8 bit extension of ASCII, which provides 193 graphical symbols of the Latin Script, covering most western european languages and standard romanizations of east asian languages

- Unicode (first published in late 1980s)

  it is a computing industry standard for consistent character encoding of world languages; it contains presently a repertoire of 137,439 graphical symbols covering 146 modern and historic languages; UTF-8, its most common implementation, uses 1 byte for the first 128 code points (ASCII characters) and up to 4 bytes for other characters.

Text processing in Portuguese - 2

Character encoding in UTF-8 supposes a character representation in one up to four bytes. It encompasses ASCII encoding as the one byte character representation class. All other classes are multibyte and follow the rules described below.

UTF-8 encoding format			
Byte 0	**Byte 1**	**Byte 2**	**Byte3**
0xxxxxxx			
110xxxxx	10xxxxxx		
1110xxxx	10xxxxxx	10xxxxxx	
11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

Text processing in Portuguese - 3

Portuguese special characters encodings					
Upper case	**UTF-8**	**ISO/IEC 8859**	**Lower case**	**UTF-8**	**ISO/IEC 8859**
á	C3A1	E1	Á	C381	C1
à	C3A0	E0	À	C380	C0
â	C3A2	E2	Â	C382	C2
ã	C3A3	E3	Ã	C383	C3
é	C3A9	E9	É	C389	C9
è	C3A8	E8	È	C388	C8
ê	C3AA	EA	Ê	C38A	CA
í	C3AD	ED	Í	C38D	CD
ì	C3AC	EC	Ì	C38C	CC
ó	C3B3	F3	Ó	C393	D3
ò	C3B2	F2	Ò	C392	D2
ô	C3B4	F4	Ô	C394	D4
õ	C3B5	F5	Õ	C395	D5
ú	C3BA	FA	Ú	C39A	DA
ù	C3B9	F9	Ù	C399	D9
ç	C3A7	E7	Ç	C387	C7

Text processing in Portuguese - 4

- the uppercase and lowercase alphabets should be treated as the same on detecting the occurrence of each letter
- in the same way, `á` – `à` – `â` – `ã` should be treated as instances of the letter `a`, `é` – `è` – `ê` should be treated as instances of the letter `e`, `í` – `ì` should be treated as instances of the letter `i`, `ó` – `ò` – `ô` – `õ` should be treated as instances of the letter `o`, `ú` – `ù` should be treated as instances of the letter `u` and `ç` should be treated as an instance of the letter `c`
- a *word* is defined as any sequence of characters, consisting of alphanumeric or underscore characters delimited by white spaces and separation or punctuation symbols

Text processing in Portuguese - 5

- a *white space* is a *space* character (`0x20`), a *tab* character (`0x9`), a *newline* character (`0xA`) or a *carriage return* character (`0xA`)
- a *separation symbol* is a *hyphen* (-), a *double quotation mark* (" `0x22` - “ `0xE2809C` - ” `0xe2809D`), a *bracket* (`[ ]`) or a *parentheses* (`( )`)
- a *punctuation symbol* is a *full point* (.), a *comma* (,), a *colon* (:), a *semicolon* (;), a *question mark* (?), an *exclamation point* (!), a *dash* (— `0xE28093`) or an *ellipsis* (… `0xE280A6`)
- the *apostrophe* (' `0x27`) and *single quotation marks* (‘ `0xE28098` - ’ `0xE28099`) are considered here to merge two words into a single one.

Determinant of a square matrix - 1

$$\boldsymbol{A} = \left\| \begin{matrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ & & \cdots & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{matrix} \right\| \quad , \qquad \text{where } a_{ij} \in \mathbb{R} \ \wedge \ n \in \mathbb{N}$$

$$det\, \boldsymbol{A} = \sum (-1)^{N(\alpha_1,\, \alpha_2,\, \dots\, \alpha_n)} \cdot a_{1\,\alpha_1}\, a_{2\,\alpha_2} \dots a_{n\,\alpha_n}$$

where the function $N(\alpha_1,\ \alpha_2,\ \dots\ \alpha_n)$ counts the number of term inversions in the index sequence $\alpha_1,\ \alpha_2,\ \dots\ \alpha_n$

Determinant of a square matrix - 2

Computation of the determinant of a square matrix of order *n* by direct application of the definition does not give rise to a practical procedure because

- *the number of arithmetic operations increases more than exponentially with n*

$$\text{number of multiplications} \; = \; n! \cdot (n-1)$$
$$\text{number of additions} \; = \; n! - 1$$

$$\lim_{n \to \infty} \left[n! - \sqrt{2\pi n} \left(\frac{n}{e} \right)^n \right] = 0$$

- *it does not yield a simple algorithm*

  both the generation of the *n*! products, having a matrix coefficient from each row and each column, and the determination of the number of term inversions of the associated index sequence are not easy to generate in a compact way.

Determinant of a square matrix - 3

Theorem of Laplace

$$det\, \boldsymbol{A} \;=\; \sum_{j=1}^{n} \left(-1\right)^{i+j} \cdot a_{i\,j} \cdot \overline{\boldsymbol{M}}_{i\,j}$$

where the factor $\overline{\boldsymbol{M}}_{i\,j}$ represents the complementary minor of the matrix coefficient $a_{i\,j}$

Basing the design of the algorithm for the computation of the determinant on the Theorem of Laplace in the form presented above, one converts the computation of a determinant of order *n* into the computation of *n* determinants of order *n*-1 and one gets the desired result in *n* steps, since a determinant of order 1 is equal to the coefficient of the associated matrix. Thus, a very compact recursive procedure can be generated to solve the problem.

Unfortunately, the number of arithmetic operations remains the same!

Determinant of a square matrix - 4

A different approach is to explore the concept of equivalent matrices.

Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be square matrices of order n, then the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are said to be *equivalent* if and only if

$$det\,\boldsymbol{A} = det\,\boldsymbol{B} \quad .$$

Now suppose that matrix $\boldsymbol{B}$ is upper triangular.

What is the value of its determinant?

$$\boldsymbol{B} = \begin{vmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ 0 & b_{22} & \cdots & b_{2n} \\ & & \cdots & \\ 0 & 0 & \cdots & b_{nn} \end{vmatrix} = b_{11} \cdot b_{22} \cdot \cdots \cdot b_{nn} \quad .$$

Determinant of a square matrix - 5

The method used to transform a generic square matrix of order *n* into an equivalent upper triangular matrix of the same order is based on the following elementary properties of determinants

- the determinant of a square matrix where two of its columns (rows) are swapped, is equal to the symmetric of the determinant of the original matrix
- the determinant of a square matrix built from a given square matrix by replacing one of its columns (rows) by adding to it another column (row) whose elements are multiplied by a constant factor, is equal to the determinant of the original matrix.

Determinant of a square matrix - 6

Gaussian elimination

The procedure takes n-1 steps.

In step i, one considers the coefficient a_{ii} of the squared matrix of order n

- if $a_{i\,i} = 0$, then one looks for a column $j > i$ whose coefficient $a_{i\,j} \neq 0$, swaps that column with column i and signals that the value of the determinant, once computed, has to undergo a signal reversion

<u>bear in mind</u> that if all coefficients $a_{i\,j} = 0$, for $j > i$, the procedure comes to an end and the value of the determinant is zero (**why**?)

- apply the following transformation

$$a_{k\,j} \;=\; a_{k\,j} - \frac{a_{k\,i}}{a_{i\,i}} \cdot a_{i\,j} \quad , \quad \text{for } k > i \text{ and } j \geq i \quad .$$

Determinant of a square matrix - 7

Gaussian elimination (continuation)

The number of arithmetic operations is now expressed by

$$\text{number of multiplications} = \frac{n\left(n^2+2\right)}{3} - 1$$

$$\text{number of divisions} = \frac{n\left(n-1\right)}{2}$$

$$\text{number of additions} = \frac{n\left(n^2-1\right)}{3} \quad .$$