

Relatório Trabalho Programação 3

Luís Maurício - 37722
Fábio Macarrão - 41895

2020/2021

1 Estratégia

Criámos uma função que vai testar todas as combinações possíveis, começa por fazer todas as combinações de tamanho 1 e se não encontrar uma palavra ambígua vai testar com todas as combinações de tamanho 2 e assim sucessivamente até que encontre a primeira palavra codificada ambígua.

O programa correrá infinitamente (até estourar a memória) caso não existam palavras ambiguas.

2 Escolhas

2.1 Linguagem

A linguagem escolhida foi prolog. A escolha foi motivada por ambos os membros do grupo já terem realizado as atividades das aulas práticas em prolog, logo era a linguagem em que o grupo estava mais à vontade de utilizar.

3 Objectivos

3.1 eval/2

Função recursiva que recebe um array e o conjunto de caracteres e devolve todas as combinações possíveis com os mesmos.

Esta função não é da nossa autoria, foi retirada de Prolog: How to create all possible permutations with repetition give a list of objects

3.2 myPermutation/3

Função que recebe um número (tamanho das combinações a retornar) e um conjunto de caracteres e vai verificar se existem combinações desse tamanho com a função eval. Atribui à última variável a lista de todas as combinações de tamanho pretendido.

Esta função não é da nossa autoria, foi retirada de Prolog: How to create all possible permutations with repetition give a list of objects

3.3 del/3

Função recursiva que recebe dois arrays, eliminando o primeiro array do segundo mas apenas se o primeiro for prefixo do segundo.

3.4 check/5

Função que recebe 4 variáveis:

- 1^a. Todas as letras do alfabeto que já foram confirmadas não ser prefixo do código a ser testado.
- 2^a O tuplo com a letra atualmente a ser testada como prefixo do código.
- 3^a. O resto do alfabeto ainda por testar.
- 4^a. Array que representa a palavra codificada que está a ser testada contra o alfabeto.

Esta função retorna uma palavra decodificada que corresponde a uma possível leitura do código testado, no caso de não haver possíveis leituras responde *false*.

3.5 checkperms/3

Função que recebe o alfabeto e uma lista de combinações de tamanho fixo obtida chamando a função myPermutation.

Circulando nessa lista, utiliza o *findall/3* para criar uma lista de possíveis leituras do código atualmente a ser testado, e se essa lista tiver pelo menos 2 elementos retorna uma lista com o código ambíguo e 2 possíveis leituras desse código.

3.6 start/3

Função recebe um número e o alfabeto. Com o resultado da função myPermutation chama a função checkperms, caso não haja nenhum código ambíguo na lista tenta todas as combinações com mais um caracter de comprimento.

3.7 ambiguo/4

Função que recebe apenas o alfabeto. Com esse alfabeto chama a função start/3 com o tamanho 1. Retorna o código menor que seja ambíguo e duas possíveis interpretações desse código.