Trabalho Prático de Programação III

João Cavaco
nº l42470@alunos.uevora.pt

22 de janeiro de 2021

Resumo

Este trabalho tem como objectivo escrever um programa que receba um código ambíguo e devolva a mensagem codificada mais curta (caso haja várias, a primeira por ordem lexicográfica) e duas das suas possíveis interpretações em PROLOG (SWI-Prolog).

1 Organização do código

O código está organizado da seguinte forma:

- Para armazenar os dados recebidos no input como factos para que possam ser acedidos posteriormente foram utilizados os seguintes predicados:
  - `store_code/1`
  - `store_code/2`
  - `dynamic code/2`
- Para obter as permutações de N elementos foram usados os seguintes predicados:
  - `get_permutations/2`
  - Devido ao facto de não ter conseguido implementar um predicado para o cálculo de permutações com repetição de elementos e para conjuntos de diferentes tamanhos, e devido ao facto de que os predicados incluídos no SWI-Prolog para o cálculo de permutações apenas permitem calcular permutações de N elementos em conjuntos de N elementos recorri ao uso dos seguintes predicados que se encontram dísponiveis neste artigo.
    - `nPIr/3`
    - `pisub/4`
    - `addx/4`
- Para obter todos os códigos ambíguos foram os usados os seguintes predicados:
  - `get_ambiguous_codes/2`
  - `code_to_char/2`
- Para obter todas mensages ambíguas e a mensagem mais curta foram os usados os seguintes predicados:
  - `get_ambiguous_messages/1`
  - `ambiguo/4`

2 Resolução do problema

O programa:

1. Começa por apagar todos os factos para o predicado dinâmico code/2 no ficheiro.
2. Guarda cada elemento da lista de códigos como um facto usando um termo complexo com o formato code(Letra,Codigo).
3. Calcula todas as permutações com o número de factos code/2 armazenados em conjuntos de 2 a 6.
4. Calcula todos os códigos que podem ser interpretados de forma ambígua para cada uma da listas de conjunto de permutações.
5. Calcula todos as mensagens para todos os códigos do passo anterior.
6. Calcula o tamanho de cada código e agrupa a informação numa lista que contém pares em que cada chave é o tamanho do código e o valor é o código e as suas duas interpretações.
7. Ordena a lista obtida no passo anterior e selecciona o par com o código com menos comprimento.
8. Por fim, unifica as variáveis M, T1 e T2 com os valores do par obtido no passo anterior.

3 Limites de funcionamento do programa

- O programa apenas consegue devolver uma mensagem codificada e as suas duas possíveis interpretações se uma das interpretações for uma letra e a outra interpretação for uma combinação de 2 ou mais letras.
- Para o predicado get_permutations/2 a variável N não pode tomar valores superiores a 6 pois leva a `ERROR: Out of global stack`.

4 Bibliografia

- Logic programming for combinatorial problems
- https://www.swi-prolog.org/pldoc/doc/_SWI_/library/lists.pl
- https://www.swi-prolog.org/pldoc/doc/_SWI_/library/pairs.pl
- http://www.learnprolognow.org/lpnpage.php?pageid=online
- Slides disponibilizados pelos docentes da cadeira.