

Língua Natural

1º Mini-Projeto

- Alexandre Machado nº81183,
- Filipe Azevedo nº82468,
- Grupo nº 36

Para evitar repetições nos transdutores decidimos criar 4 transdutores muito simples que apenas escrevem na saída o mesmo que receberam de entrada (com a ajuda do script “comp.py”, que também nos permite utilizar expressões regulares): um transdutor para as letras ‘a-z’ (de agora em diante denominado transdutor “lt”), um transdutor para os números ‘0-9’ (transdutor “dt”), um transdutor para as letras ‘IVXLC’ (que correspondem às letras necessárias para construir números romanos de ‘1-100’) (transdutor “rnt”), e um transdutor apenas para o símbolo ‘_’ (transdutor “wst”).

De forma a construir o Transdutor Romanos, nós optamos por construir primeiro 3 transdutores mais simples: o primeiro transforma apenas as unidades, o segundo transforma as dezenas e o terceiro transforma apenas o número ‘100’. Basta depois fazer a concatenação dos transdutores das dezenas e das unidades, de modo a obter um transdutor que traduz qualquer número entre ‘10’ e ‘99’, seguido da união deste novo transdutor com o transdutor das unidades e o transdutor do número ‘100’, evitando assim repetições desnecessárias.

Para o Transdutor 1 bastou apenas fazer uma série de concatenações e uniões do Transdutor Romanos e dos transdutores auxiliares “wst”, “rnt” e “lt” referidos acima, como se pode verificar na linha 158 do script “run.sh”.

Em relação ao Transdutor 2 bastou-nos apenas criar um transdutor com a chave de transformação para as letras ‘IVXLC’ e, tal como no Transdutor 1, concatenar, unir e ainda utilizar a operação de fecho sobre este transdutor e sobre os transdutores “dt”, “lt” e “wst”, como se pode verificar na linha 163 do script “run.sh”.

E, por último, o Transdutor 3 foi criado também através de concatenações, uniões e operações de fecho, tal como o Transdutor 2, sobre os transdutores “dt”, “rnt”, “wst” e sobre um transdutor simples criado por nós que utiliza a chave dada no enunciado para converter algumas letras (linha 168 do script “run.sh”). Foram utilizados pesos neste transdutor criado por nós de forma a resolver o caso das letras ‘m’ e ‘i’ que têm uma tradução diferente caso precedam as letras ‘b’ e ‘r’, respectivamente.

Os Transdutores Codificador e Decodificador são obtidos pela composição dos transdutores 1, 2 e 3, e dos transdutores 3, 2 e 1, respectivamente linhas 172 e 176 do script “run.sh”, tal como está definido no enunciado.

Os grafos das mensagens a codificar/descodificar encontram-se no directório graphs/words/output e os grafos dos transdutores encontram-se no directório graphs/core.