Grupo 63 Elementos: 73008, Duarte Pompeu

Estrutura do projecto

Não tendo sido especificada a hierarquia de pastas, distribui os ficheiros por pastas. Os nomes dos ficheiros respeitam as regras do enunciado e os nomes das pastas são intuitivos: romanos/, 1/, 2/, 3/, codificador/, descodificador/. Dentro destas pastas, estão os txt, fst e pdf's. Os pdf's pedidos têm nome equivalente aos FST pedidos (mas com extensão PDF). Os mails descodificados/codificados encontram-se na pasta descodificador ou codificador, e o nome é a ordem pela qual aparecem: descodificador/mail{1,2}.pdf, codificador/mail1.pdf.

Scripts

aux.sh define várias funções auxiliares que usei, para evitar a verbosidade dos comandos openfst. run.sh compila todos os ficheiros para .fst e .pdf. Também converte alguns ficheiros compactos para código fonte do openFST, usando o script word2fst. Incluo o material de apoio na entrega devido a ter os scripts python e o ficheiro syms.txt.

Desenvolvimento

O projecto foi relativamente simples, a parte mais difícil foi instalar o openFST, aprender os seus comandos e testar diferentes entradas e saídas.

Testes

Foram testados todos os exemplos do enunciado com sucesso, usando word2fst e composição de FSTs.

Problemas

Na descodificação do mail, aparece uma ambiguidade errada, pois entendo que a descodificação deveria ser linear. Esta ambiguidade é: eps -> eps eps -> 0

Isto deve-se à inversão do 1º transdutor, que usa o FST "romano". Assumi que uma entrada do tipo '09' era possível, sendo convertida para 0:eps->9/I->eps:X Tal como assumi que podia não haver decimal, por exemplo na entrada '9', e seria preciso uma transição eps:eps->9/I->eps:X. Assim, no inverso, surge a ambiguidade eps:eps/eps:0. Como não vejo maneira de desambiguar este caso, deixo-o assim, com a nota a explicar o raciocínio.