

Autores: Alexandre Antunes Rodrigues **89404 (50%)**;

João Luís Ambrósio Limeiro **98702 (50%)**;

Ambos trabalhamos de igual forma visto que a maior parte do projeto foi feito no técnico e, por conseguinte, todos os exercícios tiveram a participação de ambos.

Este mini-projeto tem como objetivo a resolução dos exercícios propostos com a criação de transdutores que são máquinas de estados finitos que recebem uma fita de entrada e originam uma fita de saída. Os transdutores criados no exercício 1 foram testados (*Test_NomeTransdutor*), assim como foi originado um transducer final resultante do teste (*ResNomeTransdutor*). Para o exercício 2, nos testes colocamos o nome *nmecNomeTransdutor_Test* e no resultado *nmecNomeTransdutor*.

Descrição dos exercícios resolvidos:

Exercício 1:

mm2mmm) Ligando dois estados ao estado inicial, foram criadas duas transições em que todos os números com apenas **um** dígito, ou seja, a começar com o 0 (ex: 01) são separados dos com **dois** dígitos, isto é, a começar com o 1 (ex: 22).

d2dd) A cada numeral árabe (1-9) foi acrescentado ao output um “0” à esquerda; Os restantes números com mais de 2 dígitos são aceites pelo transdutor gerado e são inalteráveis.

d2dddd) Em vez de gerarmos todos os caminhos possíveis, definimos todas as transições certas deixando o transdutor escolher o menor caminho e com o menor peso, para garantir que adicionamos os 0's à esquerda. Os pesos dos “*eps:0*” vão aumentado de nó para nó.

copy) Como apenas era para aceitar um único símbolo, ao aceitar um vai para o estado final, caso tenha mais símbolos na fita de entrada vai para um estado não final.

skip) Mesmo raciocínio do exercício 1.d diferindo apenas na fita de saída que é “eps”.

date2year) A fita de entrada é lida, gerado como fita de saída para cada símbolo antes do ano “eps”, sendo que no final quando começa a ler o ano começa a dar como output cada dígito de igual valor;

leap) A única informação que usamos são os últimos dois dígitos do ano, sendo que para os anos **leap**, se o penúltimo dígito for **par** e o último dígito tem de ser 0, 4 ou 8, caso seja **ímpar** deverá ser 2 ou 6 o resto dos anos são **not-leap**.

R2A) O desafio foi dar uma fita de entrada do numeral árabe 0 ao 99 e traduzir para numeração romana. A partir daí as transições entre estados tornam-se fáceis de resolver, visto que é só acrescentar à esquerda o símbolo correspondente ao numeral árabe pretendido (100-3999).

Exercício 2:

A2R) Modificamos o R2A com a função *fstinvert*.

birthR2A) Usamos o transdutor R2A.fst e o copy.fst definidos no exercício anterior, o R2A foi útil para converter os números romanos em números árabes enquanto o copy foi útil para copiar a “/”.

birthA2T) Do mesmo modo que solucionamos a alínea anterior, mas em vez de usar o R2A.fst para converter os números usamos o d2dd.fst para copiar o dia, o copy.fst com a mesma finalidade, o mm2mmm.fst para converter o mês por extenso e o d2dddd.fst para copiar o ano.

birthT2R) O primeiro passo foi inverter o d2dd, para remover o 0 à esquerda dos números, depois invertimos o mm2mmm para converter o mês por extenso em número e de seguida usamos o A2R usado na primeira alínea do exercício 2.

birthR2L) Usamos o birthR2A depois usamos o date2year.fst para saber apenas o ano e por fim o leap.fst tudo com compose.