

Report MP2 Natural Language

Taíssa Ribeiro – 86514
Madalena Galrinho - 87546

1. Descrição das opções tomadas e comentários da solução desenvolvida

text2num: Visto que o objetivo é apenas juntar as horas com os minutos no formato "HH:MM", decidimos usar apenas o *horas.fst* e o *minutos.fst*, acrescentando um novo transducer (que nomeámos de *e2dot.fst*) capaz de transformar "e" em ":". Desta forma, concatenamos primeiro o *horas* e o *e2dot*, e apenas de seguida o *minutos*. Esta concatenação apenas liga os estados finais do primeiro transducer ao estado inicial do segundo.

lazy2num: Aqui o objetivo é conseguir aceitar apenas horas em escrito e colocá-las no formato "HH:00", sendo também capaz de aceitar todos os inputs do *text2num*. Para isso criámos um novo transducer (chamado *zeros.fst*), cujo output é ":00", sem receber qualquer input, ou seja epsilon. Adicionalmente, concatenámos o *e2dot* e o *minutos*, para depois ser possível fazer a união desta concatenação com o novo transducer *zeros*. Finalmente, concatenámos o *horas* com o resultado da união referida. Desta forma, se o transducer receber como input as horas seguido de um "e", também irá fazer a tradução dos minutos, caso contrário, se receber apenas as horas em escrito, acrescenta ":00" depois de "HH".

rich2text: Este transducer deve aceitar como input horas, seguido de minutos, que se encontram no formato de "meia" ou "quartos". O seu output deve pôr por extenso a conversão do "meia" e dos "quartos" no número de minutos correspondente. Os transducers *meia.fst* e *quartos.fst* fazem exatamente essa conversão, pelo que apenas temos que permitir o reconhecimento das horas antes dos minutos. Para isso, concatenamos o *horas* e o *e2dot*, projetando o input, o que põe o output igual ao input. De seguida, concatenamos este último à união entre o *meias* e *quartos*, de forma a conseguir aceitar as duas possibilidades.

rich2num: Visto que o propósito aqui era conseguir aceitar todos os inputs anteriores percebemos que teríamos de unir os transducers anteriores. Para tal, usámos o método *compose* com o *rich2text* e o *text2num*, de forma a fazer a conversão para números. De seguida, fizemos a união com o *lazy2num* para conseguir aceitar as diferentes possibilidades, como por exemplo receber apenas horas.

num2text: Neste último transducer o objetivo é exatamente o contrário do *text2num*, ou seja, queremos aceitar as horas no formato "HH:MM" e pô-lo por extenso incluindo a palavra hora(s) e minuto(s). Sendo que apenas queremos o inverso do *text2num* fizemos exatamente isso, invertemos esse transducer, assim tudo o que era output passou a ser input e vice-versa. Finalmente, tendo em conta que a(s) hora(s) e o(s) minuto(s) têm que aparecer obrigatoriamente, recorreremos ao uso de pesos nos transducers *horas* e *minutos*. Nestes dois foi acrescentado um peso maior caso as palavras hora(s) ou minuto(s) não apareçam, quando não invertidos esse peso não os afeta, mas caso contrário dá prioridade ao aparecimento dos referidos anteriormente. Por exemplo, vendo as transições (eps, eps, 1) e (eps, horas) a última seria escolhida por ter menor peso (0).

2. Participação

Taíssa Ribeiro – 50%, Madalena Galrinho – 50%

Ambos os membros do grupo contribuíram igualmente para a realização do projeto.