MP2 Transducers - Time Conversion

Michael Madeira
Instituto Superior Técnico
Lisbon, Portugal
94129

DESCRIPTION OF TRANSDUCERS

To create a module that could couple with a speech recognition, in order to convert time from text to numerical format, numerical to text and with some variants, it is first built 4 main transducers:

- *horas* accepts text as input and outputs the left numerical part of the format *hh:mm*
- *minutos* accepts text as input and outputs the right numerical part of the format *hh:mm*
- *meias* accepts text "meia" as input and outputs "trinta" as text
- *quartos* accepts the three text variants text "um quarto", "dois quartos", "tres quartos", and outputs "quinze", "trinta", "quarenta e cinco" respectively.

There are many ways to mix and edit the latter transducers, but the main results of this module can be in raw text, processed text and numbers format, given the input number format or text, so 5 more transducers were built:

- *text2num* receives the complete text format *X horas e Y minutos* and returns the complete number format *hh:mm*
- *lazy2num* can receive the complete text format *X horas e Y minutos* but in addition it also accepts inputs *X horas* or only hours *X* characterized as lazyness, and returns the complete number format *hh:mm* as well
- *rich2text* accepts the normal text format in the hours part, but the minutes part only receives rich text variations mentioned above like *X horas e dois quartos* and it returns the the output of *hours* transducer with *quartos* output
- *rich2num* is equal to latter one but now it returns in number format *hh:mm* and it receives the lazy input
- *num2text* is totally the opposite of *text2num*

OPERATIONS AND TRANSFORMATIONS CHOSEN

The transducer *text2num* starts a dependency of a auxiliary connection between the transducers *horas* and *minutos*. This auxiliary transducer only needed two states and one transition *e::* to represent the symbol ":". And finally to connect the three transducers a concatenation is required between *horas* and *aux_e* and next with *minutos*.

The lazy way to say time only added a new transition *eps::* to the previous auxiliar transducer, since *eps* defines that the current input state will not be read, but it writes ":". This way the transducer can now accept inputs like "dez" or "oito" as o'clock hours and it is returned "10:00" and "08:00" respectively.

On transducer *num2text*, weights have to be implemented on states that were inverted in order to receive numbers as input and output text, because the input of the possible transitions turns out to be the same (e.g. 3:horas; 3:eps). To fulfill the objective of *num2text* the words *horas* e *minutos* have to be in the output, so the transition *3:horas* have to be chosen with priority over *3:eps*, by applying more weight (representing cost to do a transition) to the unwanted one, like 0.2 to *3:horas* and 0.5 to *3:eps*. The operations applied here were three inverts, to *horas*, *minutos* and the auxiliary *aux_e*, then two concatenations.

For *rich2text* there were implemented two projections, of *horas* and *aux_e* concerning the output text required, since project operation copies each arc's input label to its output label. Then the rich text "raw" transducers *meias* and *quartos* could be the output result, it is made an union of the two. Then to finalize it is concatenated all 3 transducers *horas* and *aux_e* with *meias_e_quartos*.

Finally to build *rich2num* two compositions of *meias* and *quartos* with *minutos* must be made, to transform the output of the rich text "raw" transducers, the same of the *minutos* transducer where the input is equally shared, for the sake of the numeric format output desired. Then two unions are applied to join *meias_composed* and *quartos_composed* with *minutos*. To finish it is done a simple concatenation of *horas_e_eps* with the union formed before.