

Regensburg, Sep. 2011

## Regular expressions for ELAN users

Ulrike Mosel

umosel@linguistik.uni-kiel.de

### Symbols

Table 1

| symbol   | place                                       | meaning  |
|----------|---|--|
| \b       | at the beginning and/or the end of a string | word boundary                                  |
| \w+      | at the end of a string                      | variable end of word                           |
| .        | anywhere                                    | any letter                                     |
| .*       | between spaces                              | any string of letters between spaces/ any word |
| .*\      | between spaces                              | any string of words                            |
| (x y)    | anywhere                                    | either x or y                                  |
| [^x]     | place at the beginning                      | not x  |
| (....)\1 | anywhere                                    | words with four reduplicated letters           |
| ?        | after a letter                              | preceding letter is optional                   |

### Search for particular word forms

Table 2: Combine symbols to find words with particular beginnings, endings and reduplications

| symbols                  | hits  | examples  |
|--------------------------|---|---|
| sa                       | all words containing the string <i>sa</i>                                   | <i>sa, vasaku, sahata, tisa</i>                     |
| \bsa                     | all words starting with <i>sa</i>   | <i>sa, sahata, sana, NOT vasaku, tisa</i>           |
| \bsa\b                   | all words <i>sa</i>   | <i>sa</i>   |
| \bsa.\b                  | all words consisting of <i>sa</i> and two letters that follow <i>sa</i>     | <i>saka, saku, sana,</i>                            |
| \bsa\w+                  | all words beginning with <i>sa</i> , but not <i>sa</i> by itself            | <i>sahata, sana</i>                                 |
| \b.*ana\b                | all words ending in <i>ana</i>  | <i>sinana, tamuana, sana, bana, maana</i>           |
| \b[^(bana maana)].*ana\b | all words ending in <i>ana</i> , but not <i>bana</i> or <i>maana</i>        | <i>sinana, tamuana, sana</i>                        |
| (....)\1                 | all words with four reduplicated letters                                    | <i>pakupaku, vapakupaku, mahumahun, vamahumahun</i> |
| \b(....)\1               | all words beginning with four reduplicated letters                          | <i>pakupaku</i><br>NOT: <i>vapakupaku</i>           |
| \b(....)\1ana\b          | all words beginning with four reduplicated letters and ending in <i>ana</i> | <i>vasuvasuana, hunuhunuana</i>                     |
| \bva(....)\1             | all words with the prefix <i>va-</i> and                                    | <i>vapakupaku, vagunagunaha</i>                     |

Regensburg, Sep. 2011

|            |  |                              |
|------------|--|------------------------------|
|            | four reduplicated letters                  |                              |
| \bvahaa?\b | all tokens of <i>vahaa</i> and <i>vaha</i> | <i>vahaa</i> and <i>vaha</i> |

### Searching for particular sequences of words

Table 3: Combine: \b, .\* \w+ and (x|y)

|    | symbols                         | hits  | examples  |
|----|---------------------------------|---|---|
| 1. | \bsaka\b .* \bhaa               | string of 3 words:<br>(1) <i>saka</i><br>(2) any word, and<br>(3) the word <i>haa</i> by itself or with suffixes  | <i>saka antee haa</i> ;<br><i>saka abana haari</i> ;<br><i>saka kabuu haana</i> |
| 2. | saka .* \bhaa\w+                | string of 3 words:<br>(1) <i>saka</i><br>(2) any word, and<br>(3) a words beginning with <i>haa</i> , but not <i>haa</i> by itself                                    | <i>saka abana haari</i> ;<br><i>saka kabuu haana</i>                            |
| 3. | (\bsaka\b \bsa\b) \bpaku\b      | all 2 word strings that consist of <i>saka</i> or <i>sa</i> and <i>paku</i>   | <i>saka paku</i> , <i>sa paku</i>   |
| 4. | (\bsaka\b \bsa\b) .* \bvaha\b   | all 3 word strings with<br>(1) <i>saka</i> or <i>sa</i> ,<br>(2) any word<br>(3) <i>vaha</i>  | <i>saka tii vaha</i><br><i>sa tapaku vaha</i>                                   |
| 5. | (\bsaka\b \bsa\b) (...)\1 \bhaa | all 3 word strings with<br>(1) <i>saka</i> or <i>sa</i> ,<br>(2) a word with four reduplicated letters<br>(3) the word <i>haa</i> or a word beginning with <i>haa</i> | <i>sa natanata haa</i> ,<br><i>saka natanata haana</i>                          |

#### Comments on Table 4:

*saka/sa ... haa* is a discontinuous negation. The last component *haa* can have a suffix that indicates imperfective aspect and person, e.g. *haana*, *haari*, *haara*. The formulars above provide data for the following questions:

1. Which words are used inbetween *saka* and *haa/haana/haari/haara* ?
2. Which words are used inbetween *saka* and *haana/haari/haara* ?
3. Are there examples for *saka/sa* followed by *paku* ‘do’?
4. Which words are used between *saka/sa* and *vaha* ‘back, also, again, anymore’?
5. Does *saka/sa ... haa* combine with reduplicated words?

Regensburg, Sep. 2011

## Multilayer search with regular expressions

Multilayer search is useful if you want to find examples for one meaning of polysemous or homonymous lexical items. For example, *beera* means ‘big’ and ‘chief, chiefs and chiefly’. If I want to search only for the second meaning, I use multilayer search on the transcription (t) and the free translation tier (f):

Multilayer search is also practical, if you do not know the language well and you want to search for the lexical item and any of its translations. Then you search on the free translation tier with a wild card: .\*

Note that there are still some bugs in ELAN. In the following example you see in line 4 and 6 funny things on the right hand side. This is not the translation tier, but our ‘notes’ tier. But otherwise the hits are fine. You see that *beera* is translated by ‘older’, ‘big’ and ‘important’.

Regensburg, Sep. 2011

Find all contexts with *beera* and its translation:

transcription tier: \bbeera\b

translation tier: .\*

The screenshot shows the 'Search eaf files' application window. The interface includes tabs for 'Substring Search', 'Single Layer Search', and 'Multiple Layer Search'. The 'Domain' is set to '182 eaf files'. The 'Query History' shows a 'New Query'. The 'Mode' is set to 'case insensitive' and 'regular expression'. The search criteria are defined in a table:

| Minimal Duration | Maximal Duration | Begin After | End Before | Tier Type |
|------------------|------------------|-------------|------------|-----------|
|                  |                  | \bbeera\b   |            | t         |
|                  |                  | Overlap     |            | f         |
|                  |                  | .*          |            | f         |
|                  |                  |             |            | All Tiers |

The search results show 430 hits in 430 annotations (of 181059). The results are displayed as a list of annotations with their corresponding transcription and translation tiers:

```

#1 || |me keara beera teve paa sue,| || #2 || |and the older sister said,| || #3 || || |
#1 || |o tarai o beera.| || #2 || |the big clamshell.| || #3 || || |
#1 || |E keara beera sue vai ki bona si keara rutaa teve,| || #2 || |The older sister now said to her little sister,| || #3 || || |
#1 || |E keara beera sue vai ki bona si keara rutaa teve,| || #2 || |no 'si'| || #3 || || |
#1 || |meori paa vaanoto bono suraa o beera,| || #2 || |and they lit a big fire, | || #3 || || |
#1 || |meori paa vaanoto bono suraa o beera,| || #2 || |no 'paa'| || #3 || || |
#1 || |o toro mohina o rutaa, evehee o beera."| || #2 || |the island is small, but important."| || #3 || || |

```

The application window also shows a taskbar at the bottom with various open applications and the system clock at 9:33 AM.