Regular Expressions for Field Methods: the Kazakh Edition

Maria Gouskova

1 Introduction

- Reg(ular) Ex(pressions) are used in programming to work with strings of characters.
- A regex search looks for a string that fully or partially matches a description. Regex searches are much more powerful than search functions in PDF viewing programs such as Adobe Acrobat Reader, which can only look for contiguous matches (and usually you cannot even force them to look for exact word matches—thus, searching for `tomat` returns both "tomato" and "stomatology"). Regex syntax, on the other hand, allows you to search for any words that have the consonants "t", "m", and "t", with any number of intervening symbols, or all the lines that contain "t" and "m" but not "a", or the exact string "tomat" followed by exactly one character, and so on.
- With even a basic handle on regex, you will be able to do quick and powerful searches and test and form phonological generalizations.

1.1 Prerequisites

- This handout covers just enough regex for you to be able to search Kazakh IPA dictionary (Schott 2010). You should also consult a regex cheat sheet such as this one.
- Unless you are comfortable with the command line, you will need a Unicode-compatible text editor that can do regex searches.
  - Linux: Gedit, Kate, etc.
  - Mac OS: TextWrangler
  - Windows: Edit Pad Lite
- When you do a search on a file, make sure you are using regex search options (e.g., in TextWrangler, you would need to check "grep" —**g**(lobal)**r**(egular)**e**(xpression)**p**(rint), and probably also "wraparound", which searches the entire file and not just stuff following the cursor location).
- A note on the structure of the Kazakh dictionary file: it has xml tags, which identify each line as either an orthographic representation or a phonemic transcription. Thus, you will see each word like this:

  `<lexeme>`

  `<grapheme>тасжемiс</grapheme>`

  `<phoneme>tɑsʒemɪs</phoneme>`

  `</lexeme>`

  The brackets > and < delimit each word; you can use them in your searches to find things at word edges. You can also search for either Kazakh orthography strings—a variant of Cyrillic—or for IPA characters and strings.
- The file is in Unicode (UTF-8 encoding). If things don't look right, though, reopen the file with the correct encoding (in TextWrangler, go to `File>Reopen Using Encoding>UTF-8`). Note two quirks of the IPA transcriptions: they include ь (the Russian "soft sign", which shows that the preceding consonant is palatalized) and `w` instead of `u`. It is probably safe to replace all occurrences of `w` with `u` and to remove the ь.
- If you are working with a different file, such as the Russian paradigms list of Usachev (2004), you cannot rely on `> <` to identify word edges—rather, you need to use the special regex characters for beginnings (`^`) and ends (`$`) of lines, and spaces (`\s`). See the cheat sheet.

2 Some basic regex examples

- Here are some basic examples to get you started. There are many other things you could do; see the cheat sheet for more.
- The dot symbol . means "any one character".
  - *Example*: I want to check whether the word "fruit" was transcribed with the correct vowels, and pick up some near-minimal pairs along the way. I search for the following string:
    `>ʒ.m.s<`
    This finds:
    `ʒɑmɑs`
    `ʒemɪs` - bingo! confirmed by checking a translating dictionary
    `ʒqmɯs` - this one must be a typo in the original dictionary... it happens.
    `ʒʊmɯs`
- The wild card symbol, *, means "any number of occurrences from zero to infinity". It quantifies over the preceding character or a string bracketed with (). If you want "one or more", you would use +.
  - *Example 1*: I want to search for disharmonic words. I'll start by looking for words with the vowel [ɑ] followed somewhere downstream by the vowel [ɪ], with any number of segments intervening. I search for:
    `ɑ.*ɪ`
    This finds:
    `ɑbonementtɪk`
    `qɑterlɪ` ...and 614 more words
  - *Example 2:* I want to search for words with two non-adjacent occurrences of "r" , with a palatalized [l] between them. I search for:
    `r.*(el)+.+r`
    This finds:
    `propeller`
    `tɪrelttɪr` and 47 more words
- Anything between square brackets, `[]`, is treated as if it is separated by "or": so searching for `a[bdg]a` will turn up `aba`, `ada`, and `aga`, but not `abda`. If you want to find all instances of aCCu where both Cs are voiced stops, you would search for `a[bdg][bdg]a`.
- A more elegant version of the previous search would have been to search for `a[bdg]{2}a`. This, finds exactly 2 of whatever precedes the {} expression. Think of {digit} as a suffixing quantifier.
- Parentheses `()` enclose strings, so searching for `[æeɪiʉø](qɯ)` will locate all lines that contain the string "qɯ" immediately preceded by a front vowel.
- The symbol `^` has two meanings, depending on whether it is inside or outside the `[]` brackets. If it is inside, it means "not one of the things inside these brackets". If it is outside, it means "beginning of a line." The "not" meaning is more useful in this file. Here is an example of a search for words that have medial clusters with a sonorant preceded by any consonant: `>.*[^ɑæeɪiɯʊʉəøojw][lrmnŋ].*<`
- To locate all the monosyllables, search for: `>[ptkqʃbdgʁʒrlmnŋj]*[ɑæeɪiɯʊʉəøojw][ptkqʃbdgʁʒrlmnŋj]*<`
- Okay, I think you get the idea! Have fun, and check your results against another dictionary.

References

Schott, Kai. 2010. Ralf's kazakh dictionary. Electronic resource. URL `http://script.blau.in/kazakh-dictionary.xml.bz2`, last accessed on Dec. 19, 2011.

Usachev, Andrei. 2004. *Fully accented paradigms from Zaliznjak's (1977) grammatical dictionary.* `http://dict.buktopuha.net/all_forms.rar`.