FST tutorial: foma

L545

Dept. of Linguistics, Indiana University

Spring 2013

Finite-State Toolkits

Goal: take what we've been learning and make it more concrete

Several toolkits we could examine:

- xfst (http://www.stanford.edu/~laurik/fsmbook/home.html)
- OpenFST (http://www.openfst.org/twiki/bin/view/FST/WebHome)
- hfst (http://wiki.apertium.org/wiki/Hfst)
- SFST (http://www.ims.uni-stuttgart.de/projekte/gramotron/SOFTWARE/SFST.html)
- foma (https://code.google.com/p/foma/)

foma

"Foma is a compiler, programming language, and C library for constructing finite-state automata and transducers for various uses."

- Developed by Mans Hulden and others
- Interface is very similar to `xfst`

Installation is very easy:

- Download appropriate package: https://code.google.com/p/foma/
- If you downloaded binaries, `foma` is the program to run

Example: Syllabification

Let's start by getting a quick example overview (http://code.google.com/p/foma/wiki/ExampleScripts)

```
# toysyllabify1.script
define V [a|e|i|o|u];
define Gli [w|y];
define Liq [r|l];
define Nas [m|n];
define Obs [p|t|k|b|d|g|f|v|s|z];

define Onset (Obs) (Nas) (Liq) (Gli); # Each element is optional.
define Coda  Onset.r;                 # Is mirror image of onset.

define Syllable Onset V Coda;
regex Syllable @> ... "." || _ Syllable;
```

Tutorial

We'll work from the nice tutorial at:
http://foma.sourceforge.net/lrec2010/index.html

- I recommend downloading the handout for quick reference

0. Quick review of FSTs: `lrec1.pdf`
1. Basics of foma: `lrec2.pdf`
2. The lexicon (lexc): `lrec3.pdf`
3. Rules: `lrec4.pdf`