FST tutorial: foma

L445 / L545

Dept. of Linguistics, Indiana University
Spring 2017
Finite-State Toolkits

**Goal:** take what we’ve been learning & make it concrete

Several toolkits we could examine:

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

We’ll talk through this quick start guide to get the basics:

▶ https://code.google.com/p/foma/wiki/GettingStarted
Example: Syllabification
Import into foma with source

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:

- I recommend downloading the handout for quick reference

0. Quick review of FSTs: lrec1.pdf
2. The lexicon (lexc): lrec3.pdf
Links students have found helpful:

- https://github.com/mhulden/foma/blob/master/foma/docs/simpleintro.md
- https://fomafst.github.io/morphtut.html
- https://fomafst.github.io/regexreference.html