*Phonology* **34** (2017). Supplementary materials

# *How abstract is more abstract? Learning abstract underlying representations*

**Charlie O'Hara**
University of Southern California

## Supplementary materials

This folder contains materials for running three simulations, as well as sample results of two of these simulations – results of the third simulation are described in the paper.

Each simulation requires three files, a FirstStage.py file, which performs the phonotactic stage of learning on the relevant dataset, a SecondStage.py file, which performs morphologically aware learning and learning of underlying representations, and a data file, which has the training data. The FirstStage.py returns a set of weights meant to model the phonotactic grammar learned before a child learns alternations. This set of weights then is fed (copy-pasted in this implementation) into the SecondStage.py learner.

Three datasets are explored. All datasets have 49 verb stems, derived with a set of suffixes:

(i) allsuffixes (fullset.txt) includes all three suffix types described in the paper, /-a/, /-tk$^h$/ and /-Ca/;

(ii) a-tk (just-a-tk.txt) includes only the two most common suffixes, /-a/ and /-tk$^h$/;

(iii) Ca-tk (just-tk-Ca.txt) includes only the two suffixes which can be modelled without constraints that refer to hiatus resolution (/-tk$^h$/ and /-Ca/).

The Ca-tk simulations differ from the other two simulations, in that they do not include the constraints necessary for the hiatus resolution introduced when vowel-final stems are paired with the /-a/ suffix.

The files in results-a-tk.rtf and results-allsuffixes.rtf show the results of sample runs of the relevant simulations. The learned constraint weights are listed first, then the learned probabilities of each possible UR and the highest probability UR for each morpheme. The probability of the training data overall given the grammar learned is given at the end of each file.