

Overview of graph creation

The overall picture for decoding-graph creation is that we are constructing the graph $HCLG = H \circ C \circ L \circ G$. Here

- G is an acceptor (i.e. its input and output symbols are the same) that encodes the grammar or language model.
- L is the lexicon; its **output** symbols are **words** and its **input** symbols are **phones**.
- C represents the context-dependency: its **output** symbols are **phones** and its **input** symbols represent **context-dependent phones**, i.e. windows of N phones; see **Phonetic context windows**.
- H contains the HMM definitions; its **output** symbols represent **context-dependent phones** and its **input** symbols are **transitions-ids**, which **encode the pdf-id and other information** (see **Integer identifiers used by TransitionModel**)

This is the standard recipe. However, there are a lot of details to be filled in. We want to ensure that the output is determinized and minimized, and in order for HCLG to be determinizable we have to insert disambiguation symbols. For details on the disambiguation symbols, see below **Disambiguation symbols**.

If we were to summarize our approach on one line (and one line can't capture all the details, obviously), the line would probably as follows, where **asl**=="add-self-loops" and **rds**=="remove-disambiguation-symbols", and **H'** is **H without the self-loops**:

HCLG = asl(min(rds(det(H' o min(det(C o min(det(L o G))))))))