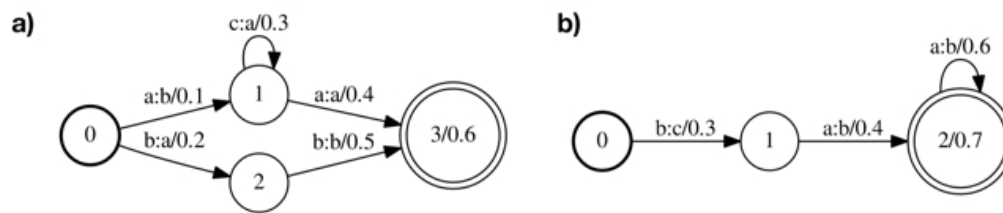


作业 1:



参考 openfst 官网,

- 1) 将图示 a)和 b)两个 WFST 写成 text 格式 a.txt.fst 和 b.txt.fst。
- 2) 定义 input label 和 output label 的字符表(即字符到数值的映射)。
- 3) 生成 a)和 b)对应的 binary 格式 WFST,记为 a.fst 和 b.fst
- 4) 进行 compose, 并输出 out.fst
- 5) 打印输出的样子。

作业 2:

从 `kaldi/src/decoder/lattice-faster-decoder.cc` 中查找

- 1) histogram pruning 的代码段
- 2) beam pruning 的代码段

作业 3:

运行 `kaldi/egs/mini_librispeech` 至少训练完 3 音素模型 tri1.

- 1) 此时你的 `data/lang_nosp_test_tglarge` 中无 G.fst 文件, 将 `data/local/lm/lm_tglarge.arpa.gz` 转化为 G.fst 存于其中。[提交你的完整命令]
- 2) 用 tri1 模型和 tgsmall 构建的 HCLG 图解码 dev_clean_2 集合的“1272-135031-0009”句, 输出 Lattice 和 CompactLattice 的文本格式。[提交你的完整命令和输出文件]
- 3) 使用 1)中生成的 tglarge 的 G.fst 和 `steps/lmrescore.sh` 对 `exp/tri1/decode_nosp_tgsmall_dev_clean_2` 中的 lattice 重打分, 汇报 wer。