# A Summary and Comparison of Two Approaches for Determinization of Lattices

Mahsa Yarmohammadi, Brian Roark, Izhak Shafran, and Richard Sproat
Center for Spoken Language Understanding (CSLU)
Oregon Health & Science University, USA

## 1 Introduction

In many applications of speech and language processing, we generate intermediate results in the form of a lattice on which we apply finite-state operations. For example, we might POS tag the words in an ASR output lattice as an intermediate stage for language modeling. Currently, we have to convert the lattice into n-best scoring sub-lattices (one sub-lattice per unique input sequence), analyze each sub-lattice separately to get its best-scoring output sequence, and combine the resulting output sequences back into a lattice. We introduce two methods to eliminate the need for this unnecessary conversion by computing *only* the 1-best scoring output sequence for *every* input sequence in the lattice. This problem arises in any finite-state tagging task such as POS tagging, word segmentation, named entity recognition, as well as in discriminative training when we need to extract the best time boundaries, acoustic, pronunciation, or language model scores in an ASR lattice.

Shafran et al. [1] and Povey et al. [4] independently proposed two solutions for the above problem. Obviously, selecting the n-best scoring sequences in the lattice is not a solution, because the result may contain more than one analysis for some input sequences while discarding all analyses of some other input sequences. Regular transducer determinization does not solve the problem either: for one thing, a transducer may include several different output sequences for a given input sequence, and still be deterministic *as a transducer*. Instead, in recently published work, Shafran et al. and Povey et al. define novel semirings in a way that determinization preserves only the best-scoring output sequence. In sections 2 and 3 we briefly introduce their methods.

In section 4, we explain our ongoing research, which is a follow-up on the previous independent projects. We are investigating the similarities and differences between the two approaches and comparing them under the same conditions.

## 2 Approach I

A semiring is 4-tuple $(R, \oplus, \otimes, \bar{0}, \bar{1})$ with a set $R$ and two binary operations such that $(R, \oplus)$ is a commutative monoid with identity $\bar{0}$; $(R, \otimes)$ is a monoid with identity $\bar{1}$; $\otimes$ distributes over $\oplus$; and $\bar{0} \otimes r = r \otimes \bar{0} = \bar{0}, \forall r \in R$. Shafran et al., encode input-output pairs as a *lexicographic* pair weight (semiring) where the first element is the original arc weight, and the second is the output symbol. In order to represent this pair as a lexicographic weight, each of the underlying semirings must obey the *path property* [2], which means that $a \oplus b$ must be either $a$ or $b$, and not some other value. The tropical semiring used to represent the original arc weights already has this property, but the standardly defined string semiring — the obvious encoding for weights representing the output symbols — does not, since it defines $a \oplus b$ as the longest common prefix of the two strings. The obvious solution of making $a \oplus b$ be the lexicographic minimum of the two strings leads to the need to explicitly represent division: during determinization, if $a \oplus b$ is, say, $a$, then what is saved as a residual (for potential future use) must be the division of $b$ by $a$, $a\backslash b$, which in turn must be *cancellative* to $b$ with the subsequent $\otimes$ operation by $a$. A natural model for this is categorial grammar [3]. For a (left) categorial semiring, we define the $\otimes$ operation as concatenation, the $\oplus$ operation as lexicographic minimum, and the $\oslash$ operation as the (left) division operation in categorial grammar.

After determinization, the deterministic lattice contains a single path per each distinct input sequence. However, the categorical weight might be accumulated in some paths, generating complex categories. We need to convert the complex categories into simple categories in order to "synchronize" the original input symbols with determinized output symbols and tropical weights. For this purpose, we compose the deterministic lattice with a *mapper* FST. The mapper FST converts complex categories into simple categories based on the notion of the *reduction* operation in categorial grammar. Shafran et al. evaluate their method on the task of POS tagging of word sequences in an ASR lattice.

# 3 Approach II

Recently, in very similar research, Daniel Povey and his colleagues proposed an alternative solution for the above described problem. Similar to our idea, they define an appropriate pair weight structure such that determinization yields to the single-best path for all unique sequences. The same as our structure, in their pair weight $(c, s)$, $c$ is the original (tropical) weight in the lattice, and $s$ is the original output symbol. The $\otimes$ operation for the semiring $s$ is concatenation, the $\oplus$ operation is the shorter length weight and then lexicographic order, and the division operation — implicitly defined in the paper — is eliminating the common prefix of the weights. Determinization and epsilon removal are done simultaneously in their method. Povey et al. evaluate their method on the task of decoding an utterance of $T$ frames to find the most likely word sequence and its corresponding state-level alignment.

# 4 Comparison

Despite large overall commonalities between the $\langle$Tropical, Categorial$\rangle$ lexicographic approach and Povey's approach, there are some interesting differences between the two which we are investigating as ongoing research. One difference is that the semiring in approach I is more complicated than approach II, because it uses structured objects with parentheses. An important difference of the approaches is *synchronization* issue. In approach I, the original input symbols are synchronized with determinized output symbols, as we mentioned earlier in section 2, whereas in approach II they are not. Approach I uses the semantics of categorial grammar to keep the history of the operations during determinizing a lattice, whereas approach II lacks this semantics. As a result, it is not easily possible to change the topology of the determinized lattice in approach II and synchronize the symbols using a mapper FST as we did in approach I. We are proposing a new algorithm — which involves consecutive symbol pushing and state splitting — to solve this issue of approach II. This algorithm changes the topology of an FST and synchronizes input and output symbols. The new algorithm is also applicable to approach I. These differences may affect time and space complexity, feasibility, and ease of use of the approaches in various tasks. We are comparing the two approaches under the same situations on the same tasks (e.g., POS tagging).

## References


[1] I. Shafran, R. Sproat, M. Yarmohammadi, and B. Roark, "Efficient determinization of tagged word lattices using categorial and lexicographic semirings," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Hawaii, 2011.

[2] M. Mohri, "Semiring frameworks and algorithms for shortest-distance problems," *Journal of Automata, Languages, and Combinatorics*, vol. 7, no. 3, pp. 321–350, Jan. 2002.

[3] J. Lambek, "The mathematics of sentence structure," *American Mathematical Monthly*, vol. 65, pp. 154–170, 1958.

[4] D. Povey, M. Hannemann, G. Boulianne, L. Burget, A. Ghoshal, M. Janda, M. Karaat, S. Kombrink, P. Motlicek, Y. Qian, K. Riedhammer, K. Vesely, and N. T. Vu, "Generating exact lattices in the WFST framework," in *IEEE Internationa Conference on Acoustic, Speech, and Signal Processing (ICASSP)*, Japan, 2012.