Language Processing with Weighted Transducers

Mehryar Mohri
AT&T Labs - Research
180 Park Avenue
Florham Park, NJ 07932-0971, USA
mohri@research.att.com

Résumé - Abstract

Les automates et transducteurs pondérés sont utilisés dans un éventail d'applications allant de la reconnaissance et synthèse automatiques de la langue à la biologie informatique. Ils fournissent un cadre commun pour la représentation des composants d'un système complexe, ce qui rend possible l'application d'algorithmes d'optimisation généraux tels que la déterminisation, l'élimination des mots vides, et la minimisation des transducteurs pondérés.

Nous donnerons un bref aperçu des progrès récents dans le traitement de la langue à l'aide d'automates et transducteurs pondérés, y compris une vue d'ensemble de la reconnaissance de la parole avec des transducteurs pondérés et des résultats algorithmiques récents dans ce domaine. Nous présenterons également de nouveaux résultats liés à l'approximation des grammaires context-free pondérées et à la reconnaissance à l'aide d'automates pondérés.

Weighted automata and transducers are used in a variety of applications ranging from automatic speech recognition and synthesis to computational biology. They give a unifying framework for the representation of the components of complex systems. This provides opportunities for the application of general optimization algorithms such as determinization, ϵ-removal and minimization of weighted transducers.

We give a brief survey of recent advances in language processing with weighted automata and transducers, including an overview of speech recognition with weighted transducers and recent algorithmic results in that field. We also present new results related to the approximation of weighted context-free grammars and language recognition with weighted automata.

Keywords: automatic speech recognition, weighted finite-state transducers, weighted automata, context-free grammars, regular approximation of CFGs, rational power series.

1 Introduction

It is a common observation that massive quantities of digitized data are widely available for various information sources such as text, speech, biological sequences, images, and handwritten

characters or patterns. But much needs to be done to fully exploit these resources since they are all highly variable or noisy sources of information. Natural language texts are extremely ambiguous, speech and handwritten texts highly variable and hard to detect in presence of noise, biological sequences often altered.

To cope with this variability, sophisticated machine learning techniques have been used to design statistical models for these information sources (Vapnik1995). The theory of weighted finite-state transducers combining the classical automata theory and statistical models provides a general framework for processing such sources of information.

Weighted transducers are in fact used in a variety of applications ranging from automatic speech recognition and synthesis to computational biology (Baldi and Brunak1998; Culik II and Kari1997; Mohri1997).

They give a common representation for the components of complex language processing systems. This provides opportunities for the application of general optimization algorithms such as determinization, ϵ-removal, and minimization of weighted transducers.

In what follows, we give a brief survey of some recent advances in language processing with weighted automata and transducers, including an overview of speech recognition with weighted transducers and recent algorithmic results in that field. We also present new results related to the approximation of weighted context-free grammars and language recognition with weighted automata.

2 Speech recognition with weighted transducers

2.1 Preliminaries

A *weighted finite-state transducer* $T = (\Sigma, \Omega, Q, E, I, F, \lambda, \rho)$ over a weight set $\mathbb{K}$ (Berstel1979; Eilenberg1974) is a generalization of the classical definition of a finite automaton and is given by a finite set of states Q, an input alphabet Σ, an output alphabet Ω, a finite set of transitions $E \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times (\Omega \cup \{\epsilon\}) \times \mathbb{K} \times Q$, an *initial state* $i \in Q$, a set of *final states* $F \subseteq Q$, an *initial weight* λ and a *final weight function* ρ. Figure 1 gives an example of weighted transducer.

For numerical stability, the weights used in speech recognition systems are often log probabilities, thus $\mathbb{K} = \mathbb{R} \cup \{\infty\}$. The weight of a path is obtained by adding the weights of its constituent transitions. We denote by $p[\pi]$ the original and by $n[\pi]$ the destination state of a path π. The weight associated by T to a pair of strings (x, y), $x \in \Sigma^*$, $y \in \Omega^*$, is then given by:

$$[\![T]\!](x, y) = \bigoplus_{P(x,y)} \lambda \cdot w[\pi] \cdot \rho(n[\pi])$$

where the sum runs over $P(x, y)$, the set of paths in T with input label x and output label y and where $\oplus$ is defined by:

$$\forall a, b \in \mathbb{R} \cup \{\infty\}, a \oplus b = -\log(exp(-a) + exp(-b))$$

When a Viterbi approximation is assumed, $\oplus$ is replaced by $\min$ in these definitions.

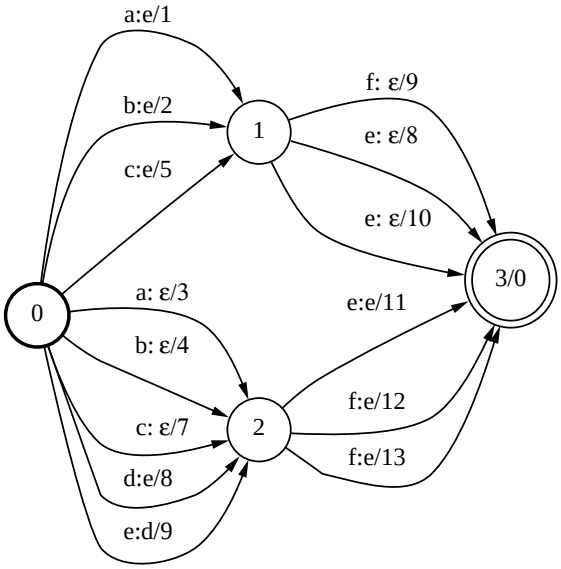

Figure 1: Example of a weighted transducer. The initial state is represented by a bold circle, final states by double circles. Inside each circle, the first number indicates the state number, the second, at final states only, the value of the final weight function ρ at that state. Each transition is labeled with $x : y/w$ where x is the input label, y the output label and w its weight. The symbol ϵ denotes the empty string.

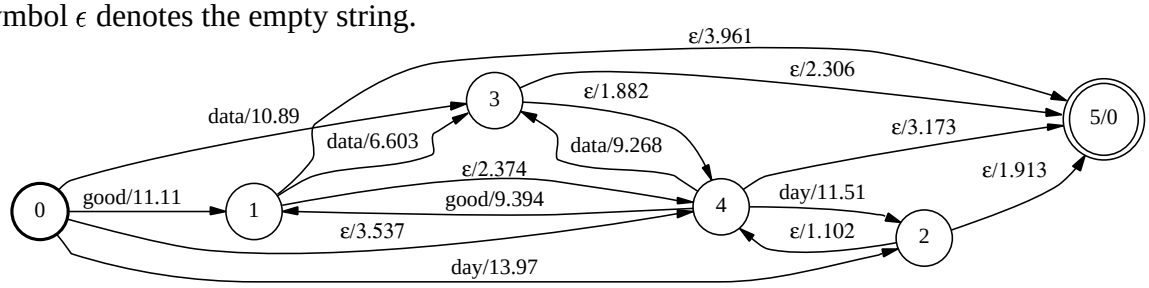

Figure 2: Toy bigram model represented by a weighted automaton.

2.2 Speech recognition components

Weighted finite-state transducers provide a common and natural representation for all the components used in the search stage of speech recognition systems: HMM models, context-dependency, pronunciation dictionaries, and language models (Mohri et al.1998). We briefly illustrate this in the following.

Most statistical grammars used in speech recognition can be represented by weighted automata. Figure 2 shows a toy grammar corresponding to an n-gram model restricted to a few words. In particular, n-gram models can be represented very naturally by weighted automata. The construction is based on associating one state to each possible $(n-1)$-gram sequence. For example, in the bigram model of figure 2, state 1 corresponds to the word "good", state 2 to the word "day" and state 3 to the word "data". More general weighted grammars such as weighted context-free grammars can also be approximated with weighted regular grammars and thus be represented by weighted automata as shown in the next section.

Pronunciation dictionaries can also be compactly represented by weighted transducers mapping

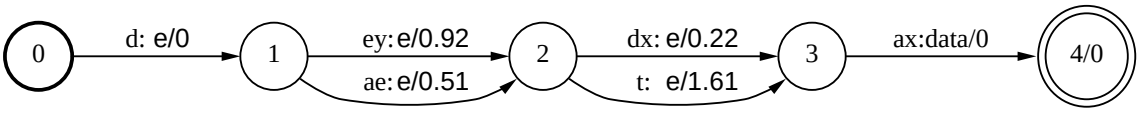

Figure 3: Sample pronunciation dictionary transducer encoding four different pronunciations of the word "data".

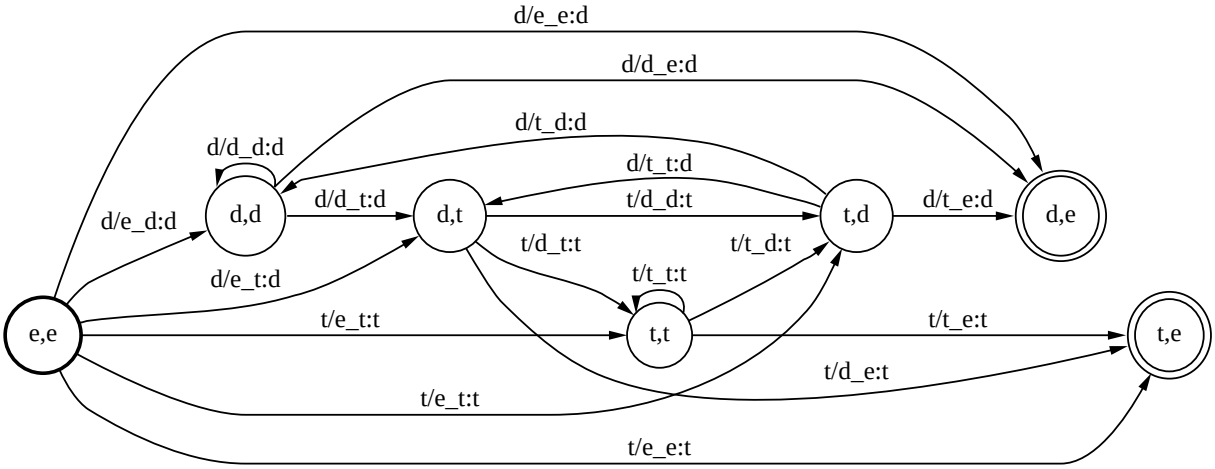

Figure 4: A triphonic context-dependent model for the phones d and t represented by a finite-state transducer. The symbol x/y_z represents the context-dependent phone x with the left context y and right context z.

phonemic transcriptions to word sequences. Figure 5 illustrates this in the particular case of the pronunciation of the word "data" in English corresponding to the four cases of flapped *d* or *t*. The weights can be used to encode the probability of each pronunciation typically based on data collected from a large number of speakers.

Figure 4 illustrates the construction of a simple triphonic context-dependency transducer C mapping context-dependent phones to phones with only two phones d and t (Pereira and Riley1997). Each state (x, y) encodes the most recent pair of phones read. e represents the start or end of a phone sequence. More general context-dependent models corresponding to weighted rewrite rules can also be compiled into weighted finite-state transducers (Mohri and Sproat1996).

Figure 5 shows a three-state HMM transducer mapping sequences of distribution indices to context-dependent phones. Such models can clearly be represented by weighted transducers. The global HMM transducer for speech recognition is obtained by taking the closure of the union of all HMMs used in acoustic modeling.

2.3 New algorithmic results

Weighted transducers map input sequences to output sequences with some weights. They can be composed like other mappings to create more complex mappings. The composition of two

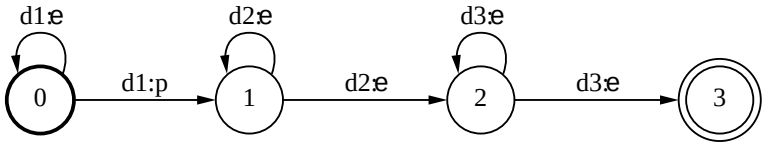

Figure 5: Three-state HMM transducer mapping sequences of distribution indices to context-dependent phones.

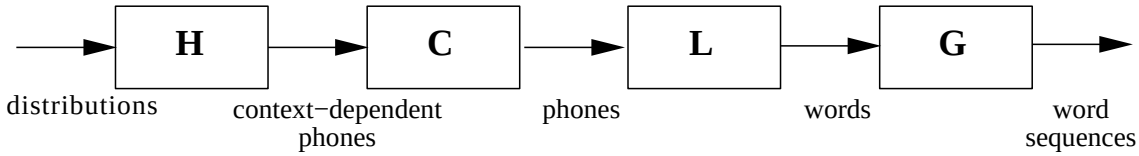

Figure 6: Recognition cascade.

transducers T_1 and T_2 is defined by:

$$\forall x, y \;\; [\![T_1 \circ T_2]\!](x, y) = \bigoplus_z [\![T_1]\!](x, z) \cdot [\![T_2]\!](z, y)$$

There exists a natural and efficient composition algorithm for combining weighted transducers (Mohri, Pereira, and Riley1996). The algorithm is an extension to the weighted case of the classical composition algorithm for unweighted transducers (Berstel1979). It is based on a filter represented by a transducer that eliminates the redundancy of ϵ-paths.

The composition of the components just presented: H, the transducer mapping sequences of distribution indices to context-dependent phone, C the context-dependent model, L the lexicon or pronunciation dictionary, and G the grammar, gives a mapping from sequences of distribution names to word sequences:

$$H \circ C \circ L \circ G$$

Figure 6 illustrates that recognition cascade. Recent work in very large-vocabulary speech recognition has shown that it is in fact possible to build off-line the result of that cascade of compositions, even for very large tasks, using general optimization algorithms such as ϵ-removal (Mohri2000a), determinization (Mohri1997) and minimization of weighted finite-state transducers (Mohri2000b).

The result is thus a single transducer that integrates all the speech recognition components, directly mapping from HMM states to words (Mohri and Riley1999). Experiments with a 463,331-word vocabulary North American Business News (NAB) Task show that this also leads to a substantial improvement of the recognition speed (Mohri and Riley1999). The size of the integrated context-dependent networks constructed can be further dramatically reduced using a factoring algorithm. With that construction, the integrated NAB recognition transducer contains only about 1.3 times as many transitions as the language model G it is constructed from (Mohri and Riley1999).

The weights of the integrated recognition transducer can be distributed in many equivalent ways along its paths. For speech recognition, the weight distribution is crucial since pruning is typically based on the combined weight from the acoustic, duration, pronunciation, and language model components accumulated so far along an explored path: the integrated recognition trans-

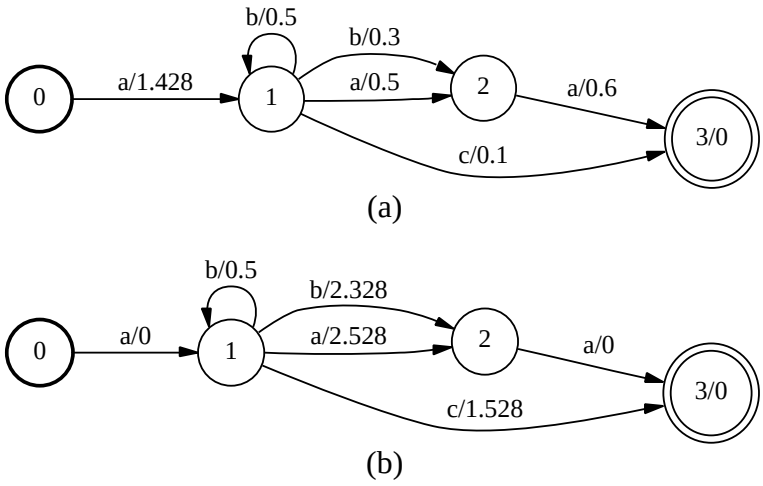

Figure 7: Weight pushing algorithm in the log semiring. The resulting automaton (b) is equivalent to (a) and is stochastic: at each state, the probability weights of outgoing transitions sum to 1.

ducer is searched with a simple Viterbi decoder combined with a beam pruning to find the best path and to output the transcription corresponding to the input speech utterance.

It is possible to modify the weights so that the sum of the probabilities for all transitions leaving a state is 1. This makes the transducer stochastic and results in an equivalent transducer whose weight distribution is more suitable for pruning and speech recognition and leads to substantial improvements in the recognition speed as demonstrated in several tasks. As an example, with this technique, we can obtain a 550% speed-up at 88% word accuracy in rescoring NAB word lattices with more accurate 2nd-pass models.

Figures 7 (a)-(b) illustrate the application of the algorithm in the case of a simple weighted automaton. There exists a generalization of the algorithm of Floyd-Warshall that can be used to effectively compute the equivalent stochastic transducer (Mohri1998). However, that algorithm has a quadratic time complexity and a cubic space complexity which makes its application to the large transducers used in speech recognition impossible in practice – such transducers may have several million transitions. A new algorithm devised recently has been shown to be practical for the computation of the resulting stochastic transducer even for such large transducers (Mohri1998). The algorithm has been shown to be practical for transducers of more than 5M transitions such as a factored integrated recognition transducer used for the 463,331-word vocabulary NAB task. It is also applied to word lattices to speed-up rescoring with more accurate 2nd-pass models.

3 Regular approximation of context-free grammars

General context-free grammars are computationally too demanding for real-time applications such as speech recognition. The grammars used in those applications often represent regular languages either by construction or as as a result of a regular approximation of a more general context-free grammar (Pereira and Wright1997; Grimley Evans1997; Johnson1998).

$$
\begin{array}{lcl}
S & \rightarrow & aB \\
S & \rightarrow & bA \\
A & \rightarrow & a \\
A & \rightarrow & aS \\
A & \rightarrow & bAA \\
B & \rightarrow & b \\
B & \rightarrow & bS \\
B & \rightarrow & aBB
\end{array}
$$

(a)

$$
\begin{array}{lcl}
S & \rightarrow & aB \\
B' & \rightarrow & S' \\
S & \rightarrow & bA \\
A' & \rightarrow & S' \\
A & \rightarrow & aA' \\
A' & \rightarrow & \epsilon \\
A & \rightarrow & aS \\
S' & \rightarrow & \epsilon
\end{array}
\qquad
\begin{array}{lcl}
S' & \rightarrow & A' \\
A & \rightarrow & bA \\
A' & \rightarrow & A \\
A' & \rightarrow & A' \\
B & \rightarrow & bB' \\
B' & \rightarrow & \epsilon \\
B & \rightarrow & bS \\
S' & \rightarrow & B'
\end{array}
\qquad
\begin{array}{lcl}
B & \rightarrow & aB \\
B' & \rightarrow & B \\
B' & \rightarrow & B'
\end{array}
$$

(b)

Figure 8: Regular approximation by transformation. (a) Context-free grammar G_1. (b) Grammar G_2 obtained from G_1 by transformation..

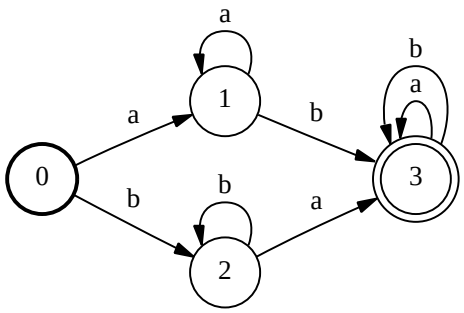

Figure 9: Finite automaton realizing the approximated grammar G_2 shown in figure 8 (b), using the compilation algorithm presented in (Mohri and Pereira1998).

The effect of such approximations are often complex and it is difficult for the grammar writer to modify the resulting grammar or to adapt it to a specific application. Furthermore, many of these approximations do not scale. They blow up for grammars of several hundred or thousand rules (Nederhof2000).

A new approximation algorithm has been devised more recently that applies to any context-free grammar and that guarantees that the result can be compiled into a finite automaton (Mohri and Nederhof2001). The resulting grammar contains at most one new nonterminal for any nonterminal symbol of the input grammar, and new rules are formed out of rules from the input grammar by means of a straightforward decomposition. The result thus remains readable and if necessary modifiable. The algorithm also extends to the case of weighted context-free grammars.

An approximate grammar G_2 is obtained from G_1 by introducing at most one new non-terminal symbol A' for each non-terminal A and by introducing the rule $A' \rightarrow \epsilon$. Each rule of the form $A \rightarrow \alpha_0\ B_1\ \alpha_1\ B_2\ \alpha_2\ \cdots\ B_m\ \alpha_m$ where $m \geq 0$ and where $B_1, \ldots, B_m$ are mutually dependent non-terminals is split into the following set of rules: $A \rightarrow \alpha_0\ B_1$, $B'_1 \rightarrow \alpha_1\ B_2$ $B'_2 \rightarrow \alpha_2\ B_3 \ldots B'_{m-1} \rightarrow \alpha_{m-1}\ B_m\ B'_m \rightarrow \alpha_m\ A'$.

Figures 8 (a)-(b) illustrate this approximation. The resulting grammar G_2 is strongly regular and can be compiled efficiently into the finite automaton of figure 9 (Mohri and Pereira1998). This approximation algorithm, as well as three other variants, have been fully implemented and incorporated in the GRM library (Mohri2001). We used that approximation algorithm and implementation to approximate a weighted grammar of about $25,000$ rules used for translation at AT&T. The transformed grammar had about $36,000$ rules. The whole approximation process including the creation of a finite automaton accepting that grammar took about one minute using

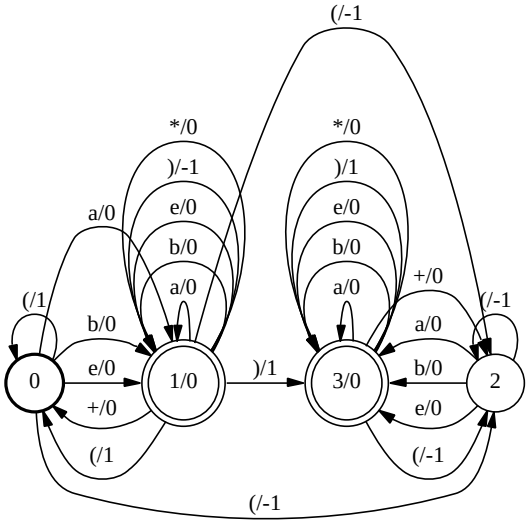

Figure 10: A 4-state weighted automaton over the tropical semiring recognizing regular expressions over the alphabet $\{a, b\}$. The symbol e corresponds to the empty string used in regular expressions. For simplicity, the empty set regular expression $\emptyset$ has been omitted.

an SGI Origin 2000.

4 Context-free recognition with weighted automata

Weighted automata can be used to recognize more complex languages than just regular languages (Cortes and Mohri2000). The definition of recognition with weighted automata is a natural generalization of that of recognition with unweighted automata. Let $\mathbb{J}$ be a subset of the weight set $\mathbb{K}$ over which the automaton has been defined. A string x is said to be $\mathbb{J}$-accepted by the automaton A when the sum [1] of the weights of all paths in A labeled with x is an element of $\mathbb{J}$. $\mathbb{J}$ is often chosen to be a singleton which makes it possible to test in constant time if a weight is in $\mathbb{J}$.

An example of a non-regular language that can be recognized by a weighted automaton is the language of regular expressions. The description of that language in formal language theory courses is often confusing since it is more powerful (it is context-free) than the set of objects it is meant to describe (regular languages). This forces the introduction of the more general concepts of context-free grammars and parsing to give a full description of the conceptually simpler regular expressions.

Figure 10 shows a simple automaton that 0-recognizes the language of regular expressions over the alphabet {a,b}. The semiring considered here is $(\mathbb{R} \cup \{\infty\}, \min, +, \infty, 0)$, the tropical semiring. Thus, a string x is accepted by that automaton iff the minimum weight of a path labeled with x is 0. This membership test can be performed in linear time.

[1] The sum here corresponds to the first operation of the semiring $\mathbb{K}$.

5 Conclusion

We gave a brief survey of recent algorithmic and theoretical results related to the use of weighted finite-state transducers in language processing. Weighted transducers provide compact representations for the components or models of language processing systems. Efficient algorithms such as composition can be used to combine these models. General optimization algorithms help reducing their size or increasing their efficiency of use. The theoretical foundation for weighted finite-state transducers, the theory of rational power series, combines the theory of probabilistic modeling and classical automata theory.

Acknowledgements

The material presented in this paper is in large parts the result of collaboration with Corinna Cortes, Mark-Jan Nederhof, Fernando Pereira, and Michael Riley.

References

Baldi, Pierre and Soren Brunak. 1998. *Bioinformatics: The Machine Learning Approach (Adaptive Computation and Machine Learning)*. MIT Press.

Berstel, Jean. 1979. *Transductions and Context-Free Languages*. Teubner Studienbucher: Stuttgart.

Cortes, Corinna and Mehryar Mohri. 2000. Context-Free Recognition with Weighted Automata. *Grammars*, 3(2-3).

Culik II, Karel and Jarkko Kari. 1997. Digital images and formal languages. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of Formal Languages*. Springer, pages 599–616.

Eilenberg, Samuel. 1974. *Automata, Languages and Machines*, volume A. Academic Press.

Grimley Evans, E. 1997. Approximating context-free grammars with a finite-state calculus. In *35th Annual Meeting of the ACL*, pages 452–459.

Johnson, M. 1998. Finite-state approximation of constraint-based grammars using left-corner grammar transforms. In *36th Annual Meeting of the ACL and 17th International Conference on Computational Linguistics*, volume 1, pages 619–623.

Mohri, Mehryar. 1997. Finite-State Transducers in Language and Speech Processing. *Computational Linguistics*, 23:2.

Mohri, Mehryar. 1998. General Algebraic Frameworks and Algorithms for Shortest-Distance Problems. Technical Memorandum 981210-10TM, AT&T Labs - Research, 62 pages.

Mohri, Mehryar. 2000a. Generic Epsilon-Removal Algorithm for Weighted Automata. In *Proceedings of the Fifth International Conference on Implementation and Application of Automata (CIAA'2000)*, London, Ontario, Canada, July.

Mohri, Mehryar. 2000b. Minimization algorithms for sequential transducers. *Theoretical Computer Science*, 234:177–201, March.

Mohri, Mehryar. 2001. Weighted Grammar Tools: the GRM Library. In Jean claude Junqua and Gertjan van Noord, editors, *Robustness in Language and Speech Technology*. Kluwer Academic Publishers, The Netherlands, pages 165–186.

Mohri, Mehryar and Mark-Jan Nederhof. 2001. Regular Approximation of Context-Free Grammars through Transformation. In Jean claude Junqua and Gertjan van Noord, editors, *Robustness in Language and Speech Technology*. Kluwer Academic Publishers, The Netherlands, pages 153–163.

Mohri, Mehryar and Fernando C. N. Pereira. 1998. Dynamic Compilation of Weighted Context-Free Grammars. In 36*th Meeting of the Association for Computational Linguistics (ACL '98), Proceedings of the Conference, Montréal, Québec, Canada*. ACL.

Mohri, Mehryar, Fernando C. N. Pereira, and Michael Riley. 1996. Weighted Automata in Text and Speech Processing. In *Proceedings of the 12th biennial European Conference on Artificial Intelligence (ECAI-96), Workshop on Extended finite state models of language, Budapest, Hungary*. ECAI.

Mohri, Mehryar and Michael Riley. 1999. Integrated Context-Dependent Networks in Very Large Vocabulary Speech Recognition. In *Proceedings of the 6th European Conference on Speech Communication and Technology (Eurospeech '99)*, Budapest, Hungary.

Mohri, Mehryar, Michael Riley, Don Hindle, Andrej Ljolje, and Fernando C. N. Pereira. 1998. Full Expansion of Context-Dependent Networks in Large Vocabulary Speech Recognition. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP '98)*, Seattle, Washington.

Mohri, Mehryar and Richard Sproat. 1996. An Efficient Compiler for Weighted Rewrite Rules. In 34*th Meeting of the Association for Computational Linguistics (ACL '96), Proceedings of the Conference, Santa Cruz, California*. ACL.

Nederhof, M.-J. 2000. Practical experiments with regular approximation of context-free languages. *Computational Linguistics*, 26(1).

Pereira, F.C.N. and R.N. Wright. 1997. Finite-state approximation of phrase-structure grammars. In E. Roche and Y. Schabes, editors, *Finite-State Language Processing*. MIT Press, pages 149–173.

Pereira, Fernando C. N. and Michael Riley, 1997. *Finite State Language Processing*, chapter Weighted Rational Transductions and their Application to Human Language Processing. The MIT Press.

Vapnik, Vladimir. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag: Berlin-New York.