On the Determinizability of Weighted Automata and Transducers

Cyril Allauzen and Mehryar Mohri
{allauzen,mohri}@research.att.com
AT&T Labs – Research
180 Park Avenue
Florham Park, NJ 07932, USA

Finite automata are classical computational devices used in a variety of large-scale applications [1]. *Finite-state transducers* are automata whose transitions are labeled with both an input and an output label. Some applications in text, speech and image processing require more general devices, *weighted automata*, to account for the variability of the input data and to rank various output hypotheses [7, 9, 8]. A *weighted automaton* is a finite automaton in which each transition is labeled with some weight in addition to the usual symbol.

Weighted automata and transducers provide a common representation for each component of a complex system used in these applications and admit general algorithms such as composition which can be used to combine these components. The time efficiency of these systems is substantially increased when *deterministic* or *subsequential* machines are used [9] and the size of these machines can be further reduced using general minimization algorithms [9, 10]. A weighted automaton or transducer is *deterministic* or *subsequential* if it has a unique initial state and if no two transitions leaving the same state share the same input label.

A general determinization algorithm for weighted automata and transducers was given by [9]. The algorithm outputs a deterministic machine equivalent to the input weighted automaton or transducer and is an extension of the classical subset construction used for unweighted finite automata. But, unlike the case of unweighted automata, not all finite-state transducers or weighted automata and transducers can be determinized using this algorithm. In fact, some machines do not admit any equivalent deterministic one. Thus, it is important to design an algorithm for testing the determinizability of finite-state transducers and weighted automata.

A characterization of determinizable finite-state transducers based on a *twins property* was given by [6, 5]. The author also proved that the twins property is decidable, see [3] for a presentation of these results in English. A polynomial-time algorithm for deciding the twins property for functional transducers was given by [11], its time complexity is $O(|Q|^4(|Q|^2 + |E|^2)|\Delta|)$ where Q is the set of states of the input transducer, E its set of transitions and Δ the output alphabet.[1]

More recently, [2] proposed a similar polynomial-time algorithm for deciding the twins property for functional transducers, its complexity is $O(|Q|^4(|Q|^2 + |E|^2))^2$ and only differs from that of [11] by the fact that it doesn't depend on the output alphabet size. We became aware of that work only recently. There are some similarities between this approach and ours, but there are some crucial technical differences that make our algorithm fundamentally different, far more practical and significantly better in terms of complexity.

We present a new, conceptually much simpler and computationally much more efficient algorithm for testing the twins property for finite-state transducers. The complexity of our algorithm is $O(|Q|^2(|Q|^2 + |E|^2))$. It is based on a general algorithm of composition of finite-state transducers and a new characterization of the twins property in terms of combinatorics of words.

[1]We are giving here the most favorable estimate of the complexity of that algorithm. The authors do not present a precise analysis of the complexity of their algorithm [11].

[2]The complexities given in [2] are wrong but they have been corrected in an extended version to appear in *Theoretical Computer Science*

In the case of weighted automata, a similar twins property was introduced by [9] to characterize the determinizability of unambiguous weighted automata over a commutative semiring. The property is also a sufficient condition for the determinizability of ambiguous machines. [9] also gave an algorithm for testing the twins property for unambiguous weighted automata. The complexity of the best existing algorithm so far for testing this property was $O(|E|^2|Q|^6)$ [4]. We present a new and efficient algorithm for testing the twins property for unambiguous and cycle-unambiguous weighted automata whose complexity is $O(|Q|^2 + |E|^2)$. We further conjecture this complexity to be optimal.

The first step of both of our algorithms, transducer and weighted automata cases, consists of constructing the intersection or composition of the input machine M and its *inverse*, $M \circ M^{-1}$, which can be done in quadratic time $O(|Q|^2 + |E|^2)$. We also give an algorithm to test the *functionality* of a transducer T, that is to determine if T represents a function, using the same composed machine $T \circ T^{-1}$. The complexity of our algorithm is $O(|Q|^2 + |E|^2)$ since it only requires work linear in the size of $T \circ T^{-1}$.

References

[1] Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman. *Compilers, Principles, Techniques and Tools*. Addison Wesley: Reading, MA, 1986.

[2] Marie-Pierre Béal, Olivier Carton, Christophe Prieur, and Jacques Sakarovitch. Squaring transducers: An efficient procedure for deciding functionality and sequentiality. In *Proceedings of LATIN'2000*, volume 1776 of *Lecture Notes in Computer Science*. Springer, 2000.

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

[4] Adam L. Buchsbaum, Raffaele Giancarlo, and Jeffery R. Westbrook. On the Determinization of Weighted Finite Automata. *SIAM Journal of Computing*, 30(5):1502–1531, 2000.

[5] Christian Choffrut. Une caractérisation des fonctions séquentielles et des fonctions sous-séquentielles en tant que relations rationnelles. *Theoretical Computer Science*, 5:325–338, 1977.

[6] Christian Choffrut. *Contributions à l'étude de quelques familles remarquables de fonctions rationnelles*. PhD thesis, (thèse de doctorat d'Etat), Université Paris 7, LITP: Paris, France, 1978.

[7] Maxime Crochemore. Transducers and Repetitions. *Theoretical Computer Science*, 45(1):63–86, 1986.

[8] Karel Culik II and Jarkko Kari. Digital Images and Formal Languages. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 599–616. Springer, 1997.

[9] Mehryar Mohri. Finite-State Transducers in Language and Speech Processing. *Computational Linguistics*, 23(2), 1997.

[10] Mehryar Mohri. Minimization Algorithms for Sequential Transducers. *Theoretical Computer Science*, 234:177–201, 2000.

[11] Andreas Weber and Reinhard Klemm. Economy of Description for Single-Valued Transducers. *Information and Computation*, 118(2):327–340, 1995.