To appear in *Proc. 25th ICALP*, Aalborg, Denmark, 1998. © Springer-Verlag 1998.

On the Determinization of Weighted Finite Automata

Adam L. Buchsbaum[1], Raffaele Giancarlo[2], and Jeffery R. Westbrook[1]

[1] AT&T Labs, 180 Park Ave., Florham Park, NJ 07932, USA.
{alb, jeffw}@research.att.com,
http://www.research.att.com/info/{alb, jeffw}.

[2] Dipartimento di Matematica ed Applicazioni, Universitá di Palermo, Via Archirafi 34, 90123 Palermo, Italy. Work supported by AT&T Labs.
raffaele@altair.math.unipa.it,
http://hpdma2.math.unipa.it/giancarl/source.html.

Abstract. We study determinization of weighted finite-state automata (WFAs), which has important applications in automatic speech recognition (ASR). We provide the first polynomial-time algorithm to test for the *twins* property, which determines if a WFA admits a deterministic equivalent. We also provide a rigorous analysis of a determinization algorithm of Mohri, with tight bounds for acyclic WFAs. Given that WFAs can expand exponentially when determinized, we explore why those used in ASR tend to *shrink*. The folklore explanation is that ASR WFAs have an acyclic, multi-partite structure. We show, however, that there exist such WFAs that always incur exponential expansion when determinized. We then introduce a class of WFAs, also with this structure, whose expansion depends on the weights: some weightings cause them to shrink, while others, including random weightings, cause them to expand exponentially. We provide experimental evidence that ASR WFAs exhibit this weight dependence. That they shrink when determinized, therefore, is a result of favorable weightings in addition to special topology.

1 Introduction

Finite-state machines and their relation to rational functions and power series have been extensively studied [2, 3, 12, 16] and widely applied in fields ranging from image compression [9–11, 14] to natural language processing [17, 18, 24, 26]. A subclass of finite-state machines, the weighted finite-state automata (WFAs), has recently assumed new importance, because WFAs provide a powerful method for manipulating models of human language in automatic speech recognition (ASR) systems [19, 20]. This new research direction also raises a number of challenging algorithmic questions [5].

A *weighted finite-state automaton (WFA)* is a nondeterministic finite automaton (NFA), A, that has both an alphabet symbol and a weight, from some set K, on each transition. Let $R = (K, \oplus, \otimes, \overline{0}, \overline{1})$ be a semiring. Then A together with R generates a partial function from strings to K: the value of an accepted string is the semiring sum over accepting paths of the semiring product of the weights along each accepting path. Such a partial function is a *rational power series* [25]. An important example in ASR is the set of WFAs with the *min-sum semiring*, $(\Re^+ \cup \{0, \infty\}, \min, +, \infty, 0)$, which compute for each accepted string the minimum cost accepting path.

In this paper, we study problems related to the determinization of WFAs. A *deterministic*, or *sequential*, WFA has at most one transition with a given input symbol out of each state. Not all rational power series can be generated by deterministic WFAs. A *determinization algorithm* takes as input a WFA and produces a deterministic WFA that generates the same rational power series, if one exists. The importance of determinization to ASR is well established [17, 19, 20].

As far as we know, Mohri [17] presented the first determinization procedure for WFAs, extending the seminal ideas of Choffrut [7, 8] and Weber and Klemm [27] regarding string-to-string transducers. Mohri gives a determinization procedure with three phases. First, A is converted to an equivalent unambiguous, trim WFA A_t, using an algorithm analogous to one for NFAs [12]. (*Unambiguous* and *trim* are defined below.) Mohri then gives an algorithm, **TT**, that determines if A_t has the *twins property* (also defined below). If A_t does not have the twins property, then there is no deterministic equivalent of A. If A_t has the twins property, a second algorithm of Mohri's, **DTA**, can be applied to A_t to yield A', a deterministic equivalent of A. Algorithm **TT** runs in $O(m^{4n^2})$ time, where m is the number of transitions and n the number of states in A_t. Algorithm **DTA** runs in time linear in the size of A'. Mohri observes that A' can be exponentially larger than A, because WFAs include classical NFAs. He gives no upper bound on the worst-case state-space expansion, however, and due to weights, the classical NFA upper bound does not apply. Finally, Mohri gives an algorithm that takes a deterministic WFA and outputs the minimum-size equivalent, deterministic WFA.

In this paper, we present several results related to the determinization of WFAs. In Section 3 we give the first polynomial-time algorithm to test whether an unambiguous, trim WFA satisfies the twins property. It runs in $O(m^2n^6)$ time. We then provide a worst-case time complexity analysis of **DTA**. The number of states in the output deterministic WFA is at most $2^{n(2\lg n+n^2\lg|\Sigma|+1)}$, where Σ is the input alphabet. If the weights are rational, this bound becomes $2^{n(2\lg n+1+\min(n^2\lg|\Sigma|,\rho))}$, where ρ is the maximum bit-size of a weight. When the input WFA is acyclic, the bound becomes $2^{n\lg|\Sigma|}$, which is tight (up to constant factors) for any alphabet size.

In Sections 4–6 we study questions motivated by the use of WFA determinization in ASR [19, 20]. Although determinization causes exponential state-space expansion in the worst case, in ASR systems the determinized WFAs are often *smaller* than the input WFAs [17]. This is fortuitous, because the performance of ASR systems depends directly on WFA size [19, 20]. We study why such size reductions occur. The folklore explanation within the ASR community credits special topology—the underlying directed graph, ignoring weights—for this phenomenon. ASR WFAs tend to be multi-partite and acyclic. Such a WFA always admits a deterministic equivalent.

In Section 4 we exhibit multi-partite, acyclic WFAs whose minimum equivalent deterministic WFAs are exponentially larger. In Section 5 we study a class of WFAs, RG, with a simple multi-partite, acyclic topology, such that in the absence of weights the deterministic equivalent is smaller. We show that for any $A \in RG$ and any $i \leq n$, there exists an assignment of weights to A such that the *minimal* equivalent deterministic WFA has $\Theta(2^{i\lg|\Sigma|})$ states. Using ideas from universal hashing, we show that similar results hold when the weights are random i-bit numbers. We call a WFA *weight-dependent* if its expansion under determinization is strongly determined by its weights.

We examined experimentally the effect of varying weights on actual WFAs from ASR applications. In Section 6 we give results of these experiments. Most of the ASR examples were weight-dependent. These experimental results together with the theory we develop show that the folklore explanation is insufficient: ASR WFAs shrink under determinization because both the topology and weighting tend to be favorable.

Some of our results help explain the nature of WFAs from the algorithmic point of view, i.e., how weights assigned to the transitions of a WFA can affect the performance of algorithms manipulating it. Others relate directly to the theory of weighted automata.

2 Definitions and Terminology

Given a semiring $(K, \oplus, \otimes, \bar{0}, \bar{1})$, a *weighted finite automaton* (WFA) is a tuple $G = (Q, \bar{q}, \Sigma, \delta, Q_f)$ such that Q is the set of states, $\bar{q} \in Q$ is the initial state, Σ is the set of symbols, $\delta \subseteq Q \times \Sigma \times K \times Q$ is the set of transitions, and $Q_f \subseteq Q$ is the set of final states. We assume throughout that $|\Sigma| > 1$. A *deterministic*, or *sequential*, WFA has at most one transition $t = (q_1, \sigma, \nu, q_2)$ for any pair (q_1, σ); a *nondeterministic* WFA can have multiple transitions on a pair (q_1, σ), differing in target state q_2. The problems examined in this paper are motivated primarily by ASR applications, which work with the *min-sum semiring*, $(\Re^+ \cup \{0, \infty\}, \min, +, \infty, 0)$. Furthermore, some of the algorithms considered use subtraction, which the min-sum semiring admits. We thus limit further discussion to the min-sum semiring.

Consider a sequence of transitions $\boldsymbol{t} = (t_1, \ldots, t_\ell)$, such that $t_i = (q_{i-1}, \sigma_i, \nu_i, q_i)$; $\boldsymbol{t}$ *induces* string $w = \sigma_1 \cdots \sigma_\ell$. String w is *accepted by* $\boldsymbol{t}$ if $q_0 = \bar{q}$ and $q_\ell \in Q_f$; w is *accepted by* G if some $\boldsymbol{t}$ accepts w. Let $c(t_i) = \nu_i$ be the *weight* of t_i. The *weight of* $\boldsymbol{t}$ is $c(\boldsymbol{t}) = \sum_{i=1}^{\ell} c(t_i)$. Let $T(w)$ be the set of all sequences of transitions that accept string w. The *weight of* w is $c(w) = \min_{\boldsymbol{t} \in T(w)} c(\boldsymbol{t})$. The *weighted language* of G is the set of weighted strings accepted by G: $L(G) = \{(w, c(w)) \mid w \text{ is accepted by } G\}$. Intuitively, the weight on a transition of G can be seen as the "confidence" one has in taking that transition. The weights need not, however, satisfy stochastic constraints, as do the probabilistic automata introduced by Rabin [22].

Fix two states q and q' and a string $v \in \Sigma^*$. Then $c(q, v, q')$ is the minimum of $c(\boldsymbol{t})$, taken over all transition sequences from q to q' generating v. We refer to $c(q, v, q')$ as the *optimal cost* of generating v from q to q'. We generally abuse notation so that $\delta(q, w)$ can represent the set of states reachable from state $q \in Q$ on string $w \in \Sigma^*$. We extend the function δ to strings in the usual way: $q' \in \delta(q, v), v \in \Sigma^+$, means that there is a sequence of transitions from q to q' generating v.

The *topology* of G, *top*(G), is the projection $\pi_{Q \times \Sigma \times Q}(\delta)$: i.e., the transitions of G without respect to the weights. We also refer to *top*(G) as the *graph* underlying G.

A WFA is *trim* if every state appears in an accepting path for some string and no transition is weighted $\bar{0}$ (∞ in the min-sum semiring). A WFA is *unambiguous* if there is exactly one accepting path for each accepted string.

Determinization of G is the problem of computing a deterministic WFA G' such that $L(G') = L(G)$, if such a G' exists. We denote the output of algorithm **DTA** by *dta*(G). We denote the minimal deterministic WFA accepting $L(G)$ by *min*(G), if one exists. We say that G *expands* if *dta*(G) has more states and/or transitions than G.

Let $n = |Q|$ and $m = |\delta|$, and let the *size* of G be $n + m$. We assume that each transition is labeled with exactly one symbol, so $|\Sigma| \leq m$. Recall that the weights of G are non-negative real numbers. Let C be the maximum weight. In the *general case*, weights are incommensurable real numbers, requiring "infinite precision." In the *integer case*, weights can be represented with $\rho = \lceil \lg C \rceil$ bits. We denote the integral range $[a, b]$ by $[a, b]_Z$. The integer case extends to the case in which the weights are rationals requiring ρ bits. We assume that in the integer and rational cases, weights are normalized to remove excess least-significant zero bits.

For our analyses, we use the RAM model of computation as follows. In the general case, we charge constant time for each arithmetic-logic operation involving weights (which are real numbers). We refer to this model as the $\Re$-RAM [21]. The relevant parameters for our analyses are n, m, and $|\Sigma|$. In the integer case, we also use a RAM, except that each arithmetic-logic operation now takes $O(\rho)$ time. We refer to this model as the $\mathcal{CO}$-RAM [1]. The relevant parameters for the analyses are n, m, $|\Sigma|$, and ρ.

3 Determinization of WFAs

3.1 An Algorithm for Testing the Twins Property

Definition 1. *Two states, q and q', of a WFA G are* twins *if $\forall (u, v) \in (\Sigma^*)^2$ such that $q \in \delta(\bar{q}, u)$, $q' \in \delta(\bar{q}, u)$, $q \in \delta(q, v)$, and $q' \in \delta(q', v)$, the following holds: $c(q, v, q) = c(q', v, q')$. G has the* twins property *if all pairs $q, q' \in Q$ are twins.*

That is, if states q and q' are reachable from $\bar{q}$ by a common string, then q and q' are twins only if any string that induces a cycle at each induces cycles of equal optimal cost. Note that two states having no cycle on a common string are twins.

Lemma 1 ([17, Lemma 2]). *Let G be a trim, unambiguous WFA. G has the twins property if and only if $\forall (u, v) \in (\Sigma^*)^2$ such that $|uv| \leq 2n^2 - 1$, the following holds: when there exist two states q and q' such that (i) $\{q, q'\} \subseteq \delta(\bar{q}, u)$, and (ii) $q \in \delta(q, v)$ and $q' \in \delta(q', v)$, then (iii) $c(q, v, q) = c(q', v, q')$ must follow.*

Definition 1 and Lemma 1 are analogous to those stated by Choffrut [7, 8] and (in different terms) by Weber and Klemm [27] to identify necessary and sufficient conditions for a string-to-string transducer to admit a sequential transducer realizing the same rational transduction. The proof techniques used for WFAs differ from those used to obtain analogous results for string-to-string transducers, however. In particular, the efficient algorithm we derive here to test a WFA for twins is not related to that of Weber and Klemm [27] for testing twins in string-to-string transducers.

We define $T_{\bar{q},\bar{q}}$, a multi-partite, acyclic, labeled, weighted graph having $2n^2$ layers, as follows. The root vertex comprises layer zero and corresponds to $(\bar{q}, \bar{q})$. For $i > 0$, given the vertices at layer $i - 1$, we obtain the vertices at layer i as follows. Let u be a vertex at layer $i-1$ corresponding to $(q_1, q_2) \in Q^2$; u is connected to u', corresponding to (q_1', q_2'), at layer i if and only if there are two distinct transitions $t = (q_1, a, c_1, q_1')$ and $t' = (q_2, a, c_2, q_2')$ in G. The arc connecting u to u' is labeled with $a \in \Sigma$ and has cost $c = c_1 - c_2$. $T_{\bar{q},\bar{q}}$ has at most $2n^4 - n^2 + 1$ vertices and $O(m^2 n^4)$ arcs.

Let $(q, q')_i$ be the vertex corresponding to $(q, q') \in Q^2$ at layer i of $T_{\bar{q},\bar{q}}$, if any. Let $RT \subseteq \{(q, q') \mid q \neq q'\}$ be the set of pairs of distinct states of G that are reachable from $(\bar{q}, \bar{q})_0$ in $T_{\bar{q},\bar{q}}$. For each $(q, q') \in RT$, define $T_{q,q'}$ analogously to $T_{\bar{q},\bar{q}}$.

Fix two distinct states q and q' of G. Let $(q, q')_{i_1}, (q, q')_{i_2}, \ldots, (q, q')_{i_s}$, $0 < i_1 < i_2 < \cdots < i_s$, be all the occurrences of (q, q') in $T_{q,q'}$, excluding $(q, q')_0$. This sequence may be empty. A symmetric sequence can be extracted from $T_{q',q}$. We refer to these sequences as the *common cycles sequences* of (q, q'). We say that q and q' satisfy the *local twins property* if and only if **(a)** their common cycles sequences are empty or **(b)** zero is the cost of (any) shortest path from $(q, q')_0$ to $(q, q')_{i_j}$ in $T_{q,q'}$ and from $(q', q)_0$ to $(q', q)_{i_j}$ in $T_{q'q}$, for all $1 \leq j \leq s$.

Lemma 2. *Let G be a trim, unambiguous WFA. G satisfies the twins property if and only if (i) RT is empty or (ii) all $(q, q') \in RT$ satisfy the local twins property.*

Proof (Sketch). We outline the proof for the sufficient condition. The only nontrivial case is when some states in RT satisfy the local twins property and their common cycles sequences are not empty. Let RT' be such a set. Assume that G does not satisfy the twins property. We derive a contradiction. Since RT' is not empty, we have that the set of pairs of states for which (i) and (ii) are satisfied in Lemma 1 is not empty. But since G does not satisfy the twins property, there must exist two states q and q' and a string $uv \in \Sigma^*$, $|uv| \leq 2n^2 - 1$, such that (i) both q and q' can be reached from the initial state of G through string u; (ii) $q \in \delta(q, v)$ and $q' \in \delta(q', v)$; and (iii) $c(q, v, q) \neq c(q', v, q')$. Without loss of generality, assume that $p = c(q, v, q) - c(q', v, q') < 0$. Now, one can show that $(q, q') \in RT'$. Then, using the fact that G is unambiguous, one can show that there is exactly one path in $T_{q,q'}$ from the root to $(q, q')_{|v|}$ with cost $p < 0$. Therefore, (q, q') cannot satisfy the local twins property.

To test whether a trim, unambiguous WFA has the twins property, we first compute $T_{\bar{q},\bar{q}}$ and the set RT. For each pair of states $(q, q') \in RT$ that has not yet been processed, we need only compute $T_{q,q'}$ and $T_{q',q}$ and their respective shortest path trees.

Theorem 1. *Let G be a trim unambiguous WFA. In the general case, whether G satisfies the twins property can be checked in $O(m^2n^6)$ time using the $\Re$-RAM. In the integer case, the bound becomes $O(\rho m^2 n^6)$ using the $\mathcal{CO}$-RAM.*

3.2 The DTA Algorithm

In this section we describe the **DTA** algorithm. We then give an upper bound on the size of the deterministic machines produced by the algorithm. The results of Section 5 below show that our upper bound is tight to within polynomial factors.

Given WFA $G = (Q, \bar{q}, \Sigma, \delta, Q_f)$, **DTA** generalizes the classic power-set construction to construct deterministic WFA G' as follows. The start state of G' is $\{(\bar{q}, 0)\}$, which forms an initial queue P. While $P \neq \emptyset$, pop state $q = \{(q_1, r_1), \ldots, (q_n, r_n)\}$ from P, where $q_i \in Q$ and $r_i \in \Re^+ \cup \{0, \infty\}$. The r_i values encode path-length information, as follows. For each $\sigma \in \Sigma$, let $\{q'_1, \ldots, q'_m\}$ be the set of states reachable by σ-transitions out of all the q_i. For $1 \leq j \leq m$, let $\rho_j = \min_{1 \leq i \leq n; (q_i, \sigma, \nu, q'_j) \in \delta} \{r_i + \nu\}$ be the minimum of the weights of σ-transitions into q'_j from the q_i plus the respective

r_i. Let $\rho = \min_{1 \le j \le m}\{\rho_j\}$. Let $q' = \{(q'_1, s_1), \ldots, (q'_m, s_m)\}$, where $s_j = \rho_j - \rho$, for $1 \le j \le m$. We add transition (q, σ, ρ, q') to G' and push q' onto P if q' is new. This is the only σ-transition out of state q, so G' is deterministic.

Let $T_G(w)$ be the set of sequences of transitions in G that accept a string $w \in \Sigma^*$; let $\boldsymbol{t}_{G'}(w)$ be the (one) sequence of transitions in G' that accepts the same string. Mohri [17] shows that $c(\boldsymbol{t}_{G'}(w)) = \min_{\boldsymbol{t} \in T_G(w)}\{c(\boldsymbol{t})\}$, and thus $L(G') = L(G)$. Moreover, let $T_G(w, q)$ be the set of sequences of transitions in G from state $\bar{q}$ to state q that induce string w. Again, let $\boldsymbol{t}_{G'}(w)$ be the (one) sequence of transitions in G' that induces the same string; $\boldsymbol{t}_{G'}(w)$ ends at some state $\{(q_1, r_1), \ldots, (q_n, r_n)\}$ in G' such that some $q_i = q$. Mohri [17] shows that $c(\boldsymbol{t}_{G'}(w)) + r_i = \min_{\boldsymbol{t} \in T_G(w,q)}\{c(\boldsymbol{t})\}$. Thus, each r_i is a *remainder* that encodes the difference between the weight of the shortest path to some state that induces w in G and the weight of the path inducing w in G'. Hence at least one remainder in each state must be zero.

3.3 Analyzing DTA

We first bound the number of states in *dta*(G), denoted $\#$*dta*(G).

Theorem 2. *If WFA G has the twins property, then $\#dta(G) < 2^{n(2\lg n + n^2 \lg|\Sigma| + 1)}$ in the general case; $\#dta(G) < 2^{n(2\lg n + 1 + \min(n^2 \lg|\Sigma|, \rho))}$ in the integer (or rational) case; and $\#dta(G) < 2^{n \lg|\Sigma|}$ if G is acyclic, independent of any assumptions on weights. The acyclic bound is tight (up to constant factors) for any alphabet.*

Proof (Sketch). Let $\tilde{R}$ be the set of remainders in *dta*(G). Let R be the set of remainders r for which the following holds: $\exists w \in \Sigma^*$, $|w| \le n^2 - 1$, and two states q_1 and q_2, such that $r = |c(\bar{q}, w, q_2) - c(\bar{q}, w, q_1)|$. The twins property implies that $\tilde{R} \subsetneq R$. In the worst case, each i-state tuple from G will appear in *dta*(G), and there are $|\tilde{R}|^i$ distinct i-tuples of remainders it can assume. (This over counts by including tuples without any zero remainders.) Therefore, $\#dta(G) \le \sum_{i=1}^{n} \binom{n}{i} |\tilde{R}|^i \le (2|\tilde{R}|)^n \le (2|R|)^n$.

General Case: Each string of length at most $n^2 - 1$ can reach a pair of (not necessarily distinct) states in G. Therefore, $|R| < n^2 |\Sigma|^{n^2}$. **Integer Case**: The remainders in R are in $[0, (n^2 - 1)C]_Z$ implying $|R| < n^2 C$; but still $|R| < n^2 |\Sigma|^{n^2}$. **Acyclic Case**: $\#dta(G)$ is bounded by the number of strings in the weighted language accepted by G, which is bounded by $|\Sigma|^n$. We discuss tightness in Section 5.

Processing each tuple of state-remainders generated by **DTA** takes $O(|\Sigma|(n + m))$ time, excluding the cost of arithmetic and $\min$ operations, yielding the following.

Theorem 3. *Let G be a WFA satisfying the twins property. In the general case,* **DTA** *takes $O(|\Sigma|(n + m)2^{n(2\lg n + n^2 \lg|\Sigma| + 1)})$ time on the $\Re$-RAM. In the (rational or) integer case,* **DTA** *takes $O(\rho|\Sigma|(n + m)2^{n(2\lg n + 1 + \min(n^2 \lg|\Sigma|, \rho))})$ time on the $\mathcal{CO}$-RAM. In the acyclic case,* **DTA** *takes $O(|\Sigma|(n + m)2^{n \lg|\Sigma|})$ time on the $\Re$-RAM and $O(\rho|\Sigma|(n + m)2^{n \lg|\Sigma|})$ time on the $\mathcal{CO}$-RAM.*

We can use the above results to generate hard instances for any determinization algorithm. A *reweighting function* (or simply *reweighting*) f is such that, when applied

to a WFA G, it preserves the topology of G but possibly changes the weights. We want to determine a reweighting f such that $\mathit{min}(f(G))$ exists and $|\mathit{min}(f(G))|$ is maximized among reweightings for which $\mathit{min}(f(G))$ exists. We restrict attention to the integer case and, without loss of generality, we assume that G is trim and unambiguous.

Theorem 2 shows that for weights to affect the growth of $\mathit{dta}(G)$, it must be that $\rho \leq n^2 \lg |\Sigma|$. Set $\rho_{max} = n^2 \lg |\Sigma|$. To find the required reweighting, we simply consider all possible reweightings of G satisfying the twins property and requiring at most ρ_{max} bits. There are $(2^{\rho_{max}})^m = 2^{m\rho_{max}}$ possible reweightings, and it takes $2^{O(n(2 \lg n + (n^2 \lg |\Sigma|)))}$ time to compute the expansion or decide that the resulting machine cannot be determinized, bounding the total time by $2^{O(n(2 \log n + (n^2 \log |\Sigma|)) + m\rho_{max})}$.

4 Hot Automata

This section provides a family of acyclic, multi-partite WFAs that are *hot*: when determinized, they expand independently of the weights on their transitions. Given some alphabet $\Sigma = \{a_1, \ldots, a_n\}$, consider the language $L = \bigcup_{i=1}^{n} (\Sigma - \{a_i\})^n$; i.e., the set of all n-length strings that do not include all symbols from Σ. It is simple to obtain an acyclic, multi-partite NFA H of poly(n) size that accepts L. It is not hard to show that the minimal DFA accepting L has $\Theta(2^{n+\lg n})$ states. Furthermore, we can construct H so that these bounds hold for a *binary* alphabet. H corresponds to a WFA with all arcs weighted identically. Since acyclic WFAs satisfy the twins property, they can always be determinized. Altering the weights can only increase the expansion. Kintala and Wotschke [15] provide a set of NFAs that produces a hierarchy of expansion factors when determinized, providing additional examples of hot WFAs.

5 Weight-Dependent Automata

In this section we study a simple family of WFAs with multi-partite, acyclic topology. We examine how various reweightings affect the size of the determinized equivalent. This family shrinks without weights, so any expansion is due to weighting. This study is related in spirit to previous works on measuring nondeterminism in finite automata [13,15]. Here, however, nondeterminism is encoded only in the weights. We first discuss the case of a binary alphabet and then generalize to arbitrary alphabets.

5.1 The Rail Graph

We denote by $RG(k)$ the *k-layer rail graph*. $RG(k)$ has $2k + 1$ vertices, denoted $\{0, T_1, B_1, \ldots, T_k, B_k\}$. There are arcs $(0, T_1, a)$, $(0, T_1, b)$ $(0, B_1, a)$, $(0, B_1, b)$, and then, for $1 \leq i < k$, arcs (T_i, T_{i+1}, a), (T_i, T_{i+1}, b), (B_i, B_{i+1}, a), and (B_i, B_{i+1}, b). See Fig. 1. $RG(k)$ is $(k + 1)$-partite and also has fixed in- and out-degrees. If we consider the strings induced by paths from 0 to either T_k or B_k, then the language of $RG(k)$ is the set of strings $L_{RG}(k) = \{a, b\}^k$. The only nondeterministic choice is at the state 0, where either the top or bottom rail may be selected. Hence a string w can be accepted by one of two paths, one following the top rail and the other the bottom rail.

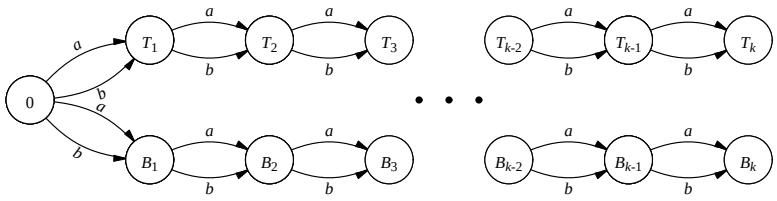

Fig. 1. Topology of the k-layer rail graph.

Technically, $RG(k)$ is ambiguous. We can disambiguate $RG(k)$ by adding transitions from T_k and B_k, each on a distinct symbol, to a new final state. Our results extend to this case. For clarity of presentation, we discuss the ambiguous rail graph.

The rail graph is weight-dependent. In Section 5.2 we provide weightings such that **DTA** produces the $(k+1)$-vertex *trivial series-parallel graph*: a graph on $k+1$ vertices, with transitions, on all symbols, only between vertices i and $i+1$, for $1 \leq i \leq k$. On the other hand, in Section 5.3 we exhibit weightings for the rail graph that cause **DTA** to produce exponential state-space expansions. We also explore the relationship between the magnitude of the weights and the amount of expansion that is possible. In Section 5.4, we show that random weightings induce the behavior of worst-case weightings. Finally, in Section 5.5 we generalize the rail graph to arbitrary alphabets.

5.2 Weighting $RG(k)$

Consider determinizing $RG(k)$ with **DTA**. The set of states reachable on any string $w = \sigma_1 \cdots \sigma_j$ of length $j \leq k$ is $\{T_j, B_j\}$. For a given weighting function c, let $c_T(w)$ denote the cost of accepting string w if the top path is taken; i.e., $c_T(w) = c(0, \sigma_1, T_1) + \sum_{i=1}^{j-1} c(T_i, \sigma_{i+1}, T_{i+1})$. Analogously define $c_B(w)$ to be the corresponding cost along the bottom path. Let $R(w)$ be the *remainder vector* for w, which is a pair of the form $(0, c_B(w) - c_T(w))$ or $(c_T(w) - c_B(w), 0)$. A state at layer $0 < i \leq k$ in the determinized WFA is labeled $(\{T_i, B_i\}/R(w))$ for any string w leading to that state. Thus, two strings w_1 and w_2 of identical length lead to distinct states in the determinized version of the rail graph if and only if $R(w_1) \neq R(w_2)$.

It is convenient simply to write $R(w) = c_T(w) - c_B(w)$. The sign of $R(w)$ then determines which of the two forms $(0, x)$ or $(x, 0)$ of the remainder vector occurs.

Let $r_i^T(\sigma)$ (rsp., $r_i^B(\sigma)$) denote the weight on the top (rsp., bottom) arc labeled σ into vertex T_i (rsp., B_i). Let $\delta_i(\sigma) = r_i^T(\sigma) - r_i^B(\sigma)$. Then $R(w) = \sum_{i=1}^{j} \delta_i(\sigma_i)$.

Theorem 4. *There is a reweighting f such that dta$(f(RG(k))) =$ min$(f(RG(k)))$, which consists of the $(k+1)$-vertex trivial series-parallel graph*

Proof. Any f for which $\delta_i(a) = \delta_i(b)$ for $i = 1$ to k suffices, since in this case $R(w_1) = R(w_2)$ for all pairs of strings $\{w_1, w_2\}$. In particular, giving zero weights suffices.

5.3 Worst-Case Weightings of $RG(k)$

Theorem 5. *For any $j \in [0, k]_Z$ there is a reweighting f such that layers 0 through j of dta$(f(RG(k)))$ form the complete binary tree on $2^{j+1} - 1$ vertices.*

Proof (Sketch). Choose any weighting such that $\delta_i(a) = 2^{i-1}$ and $\delta_i(b) = 0$ for $1 \leq i \leq j$, and let $\delta_i(a) = \delta_i(b) = 0$ for $j < i \leq k$. Consider a pair of strings w_1, w_2 of identical length such that $w_1 \neq w_2$. The weighting ensures that $R(w_1) \neq R(w_2)$.

Theorem 6. *For any $j \in [0,k]_Z$ there is a reweighting f such that layers 0 through $j-1$ of min$(f(RG(k)))$ form the complete binary tree on $2^j - 1$ vertices.*

Theorem 6, generalized by Theorem 10, shows that weight-dependence is not an artifact of **DTA** and that the acyclic bound of Theorem 2 is tight for binary alphabets.

We now address the sensitivity of the size expansion to the magnitude of the weights, arguing that exponential state-space expansion requires exponentially big weights for the rail graph. (This means that the size expansion, while exponential in the number of states, is only super-polynomial in the number of bits.)

Theorem 7. *Let f be a reweighting. If $|dta(f(RG(k)))| = \Omega(2^k)$, then $\Omega(k^2)$ bits are required to encode $f(RG(k))$.*

Proof (Sketch). There must be $\Omega(2^k)$ distinct remainders among the states at depth k in the determinized WFA, necessitating $\Omega(2^k)$ distinct permutations of the $\lceil \frac{k}{2} \rceil$ high-order bits among them. Thus $\Omega(k)$ weights must have similarly high-order bits set.

Corollary 1. *Let f be a reweighting. If $|min(f(RG(k)))| = \Omega(2^k)$, then $\Omega(k^2)$ bits are required to encode $f(RG(k))$.*

5.4 Random Weightings of $RG(k)$

Theorem 8. *Let G be $RG(k)$ weighted with numbers chosen independently and uniformly at random from $[1, 2^k - 1]_Z$. Then $E[|dta(f^R(RG(k)))|] = \Theta(2^k)$, where $E[X]$ denotes the expected value of the random variable X.*

Theorem 9. *Let G be $RG(k)$ weighted with logarithms of numbers chosen independently and uniformly at random from $[1, 2^k - 1]_Z$. Then $E[|dta(G)|] = \Theta(2^k)$.*

The proofs of Theorems 8 and 9 use the observation that the random functions defined by RG are essentially universal hash functions [6] to bound sufficiently low the probability that the remainders of two distinct strings are equal. Theorem 9 is motivated by the fact that the weights of ASR WFAs are negated log probabilities.

5.5 Extending $RG(k)$ to Arbitrary Alphabets

We can extend the rail graph to arbitrary alphabets, defining $RG(r,k)$, the *k-layer r-rail graph*, as follows. $RG(r,k)$ has $rk+1$ vertices: vertex 0 and, for $1 \leq i \leq r$ and $1 \leq j \leq k$, vertex v_j^i. Assume the alphabet is $\{1, \ldots, r\}$. $RG(r,k)$ has arcs $(0, v_1^i, s)$ for all $1 \leq i, s \leq r$ and also arcs (v_j^i, v_{j+1}^i, s) for all $1 \leq i, s \leq r$ and $1 \leq j < k$.

The subgraph induced by vertex 0 and vertices v_j^i for some i and all $1 \leq j \leq k$ comprises *rail* i of $RG(r,k)$. The subgraph induced by vertices v_j^i for all $1 \leq i \leq r$ and some j comprises *layer* j of $RG(r,k)$. Vertex 0 comprises *layer* 0 of $RG(r,k)$. Thus, $RG(2,k)$ is the k-layer rail graph, $RG(k)$, defined in Section 5.1.

Let $c(i, j, s)$ be the weight of the arc labeled s into vertex v_j^i. Theorems 4 and 5 generalize easily to the k-layer r-rail graphs. Theorem 6 generalizes to $RG(r, k)$ as follows, showing that the acyclic bound of Theorem 2 is tight for arbitrary alphabets.

Theorem 10. *For any $j \in [0, k]_Z$ there is a reweighting f such that layers 0 through $j-1$ of min$(f(RG(r, k)))$ form the complete r-ary tree on $\frac{r^j-1}{r-1}$ vertices.*

Proof (Sketch). Choose the following weighting. Set $c(i, \ell, s) = [(i + s) \mod r] \cdot r^\ell$ for all $1 \leq i, s \leq r$ and $1 \leq \ell \leq j$. Set $c(i, \ell, s) = 0$ for all $1 \leq i, s \leq r$ and $j < \ell \leq k$.

Given two strings, $w_1 \neq w_2$, such that $|w_1| = |w_2| = \ell < j$, we can show that w_1 and w_2 must lead to different vertices in *any* deterministic realization, D, of $RG(r, k)$. Assume that w_1 and w_2 lead to the same vertex in D. Let $c_d(w)$ be the cost of string w in D. Given any suffix s of length $k - \ell$, we can show that $c(w_1 s) - c(w_2 s) = c_d(w_1) - c_d(w_2)$. The right hand side is a fixed value, Δ.

Consider any position $i \leq \ell$ in which w_1 and w_2 differ. Denote the ith symbol of string w by $w(i)$. Consider two suffixes, s_1 and s_2, of length $k-\ell$, such that $s_1(j-\ell) = w_1(i)$ and $s_2(j - \ell) = w_2(i)$. Observe that the given weighting on $RG(r, k)$ forces the minimum cost path for any string with some symbol σ in position j to follow rail $(r-\sigma)$. Thus, $w_1 s_1$ and $w_2 s_1$ follow rail $r - w_1(i)$, and $w_1 s_2$ and $w_2 s_2$ follow rail $r - w_2(i)$. We can use this to show that $c(w_1 s_1) - c(w_2 s_1) \neq c(w_1 s_2) - c(w_2 s_2)$, a contradiction.

6 Experimental Observations on ASR WFAs

To determine whether ASR WFAs manifest weight dependence, we experimented on 100 WFAs generated by the AT&T speech recognizer [23], using a grammar for the Air Travel Information System (ATIS), a standard test bed [4]. Each transition was labeled with a word and weighted by the recognizer with the negated log probability of realizing that transition out of the source state; we refer to these weights as *speech weights*.

We determinized each WFA with its speech weights, with zero weights, and with weights assigned independently and uniformly at random from $[0, 2^i - 1]_Z$ (for each $0 \leq i \leq 8$). One WFA could not be determinized with speech weights due to computational limitations, and it is omitted from the data.

Figure 2(a) shows how many WFAs expanded when determinized with different weightings. Figure 2(b) classifies the 63 WFAs that expanded with at least one weighting. For each WFA, we took the weighting that produced maximal expansion. This was usually the 8-bit random weighting, although due to computational limitations we were unable to determinize some WFAs with large random weightings. The x-axis indicates the open interval within which the value $\lg(|dta(G)|/|G|)$ falls.

The utility of determinization in ASR includes the reduction in size achieved with actual speech weights. In our sample, 82 WFAs shrank when determinized. For each, we computed the value $\lg(|G|/|dta(G)|)$, and we plot the results in Fig. 2(c).

In Fig. 2(d), we examine the relationship between the value $\lg(|dta(G)|/|G|)$ and the number of bits used in random weights. We chose the ten WFAs with highest final expansion value and plotted $\lg(|dta(G)|/|G|)$ against the number of bits used. For reference the functions i^2, $2^{\sqrt{i}}$, and 2^i are plotted, where i is the number of bits. Most

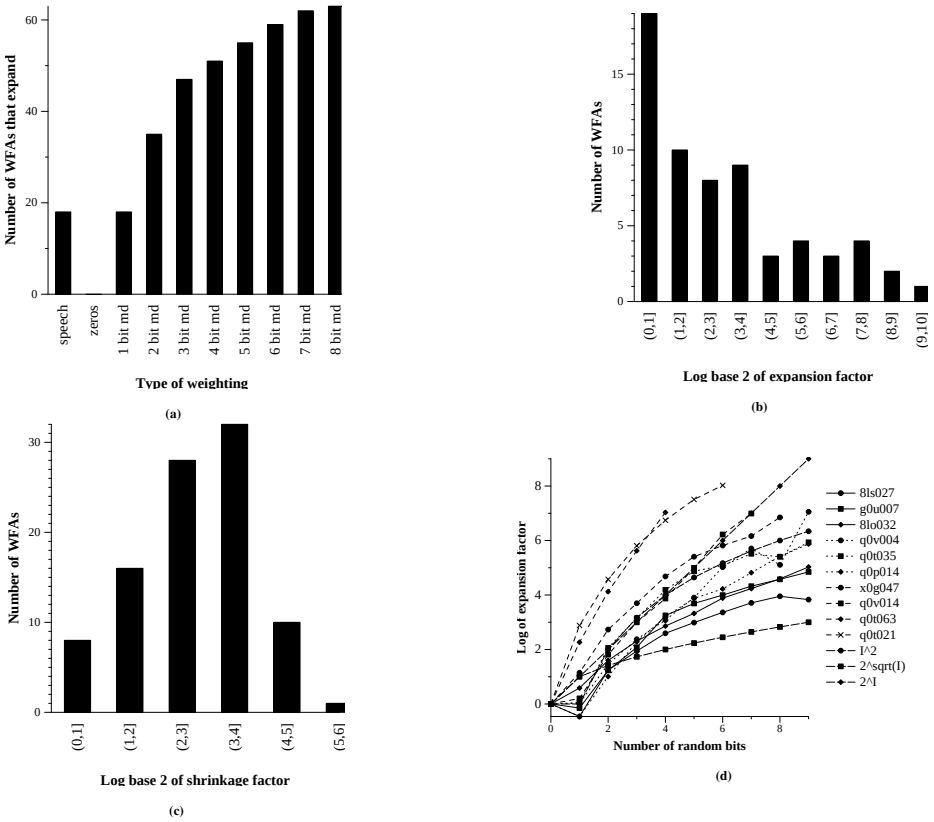

Fig. 2. Observations on ASR WFAs.

of the WFAs exhibit subexponential growth as the number of bits increases, although some, like `q0t063` have increased by 128 times even with four random bits.

The WFA that could not be determinized with speech weights was "slightly hot," in that the determinized zero-weighted variant had 2.7% more arcs than the original WFA. The remaining ninety-nine WFAs shrank with zero weights: none was hot. If one expanded, it did so due to weights rather than topology.

Figure 2(a) indicates that many of the WFAs have some degree of weight dependence. Figure 2(d) suggests that random weights are a good way to estimate the degree to which a WFA is weight dependent. Note that the expansion factor is some superlinear, possibly exponential, function of the number of random bits, suggesting that large, e.g., 32-bit, random weights should cause expansion if anything will. Analogous experiments on the minimized determinized WFAs yield results that are qualitatively the same, although fewer WFAs still expand after minimization. Hence weight dependence seems to be a fundamental property of these WFAs rather than an artifact of **DTA**.

Acknowledgements. We thank Mehryar Mohri, Fernando Pereira, and Antonio Restivo for fruitful discussions.

References

1. R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, 1993.
2. J. Berstel. *Transduction and Context-Free Languages*. Springer-Verlag, 1979.
3. J. Berstel and C. Reutenauer. *Rational Series and Their Languages*. Springer-Verlag, 1988.
4. E. Bocchieri, G. Riccardi, and J. Anantharaman. The 1994 AT&T ATIS CHRONUS recognizer. In *Proc. ARPA SLT*, pages 265–8, 1995.
5. A. L. Buchsbaum and R. Giancarlo. Algorithmic aspects in speech recognition: An introduction. *ACM J. Exp. Algs.*, 2, 1997.
6. J. L. Carter and M. N. Wegman. Universal classes of hash functions. *JCSS*, 18:143–54, 1979.
7. C. Choffrut. Une caracterisation des fonctions sequentielles et des fonctions sous-sequentielles en tant que relations rationnelles. *Theor. Comp. Sci.*, 5:325–37, 1977.
8. C. Choffrut. *Contributions á l'étude de quelques familles remarquables de function rationnelles*. PhD thesis, LITP-Université Paris 7, 1978.
9. K. Culik II and J. Karhumäki. Finite automata computing real functions. *SIAM J. Comp.*, 23(4):789–814, 1994.
10. K. Culik II and P. Rajčáni. Iterative weighted finite transductions. *Acta Inf.*, 32:681–703, 1995.
11. D. Derencourt, J. Karhumäki, M. Latteux, and A. Terlutte. On computational power of weighted finite automata. In *Proc. 17th MFCS*, volume 629 of *LNCS*, pages 236–45. Springer-Verlag, 1992.
12. S. Eilenberg. *Automata, Languages, and Machines*, volume A. Academic Press, 1974.
13. J. Goldstine, C. M. R. Kintala, and D. Wotschke. On measuring nondeterminism in regular languages. *Inf. and Comp.*, 86:179–94, 1990.
14. J. Kari and P. Fränti. Arithmetic coding of weighted finite automata. *RAIRO Inform. Th. Appl.*, 28(3-4):343–60, 1994.
15. C. M. R. Kintala and D. Wotschke. Amounts of nondeterminism in finite automata. *Acta Inf.*, 13:199–204, 1980.
16. W. Kuich and A. Salomaa. *Semirings, Automata, Languages*. Springer-Verlag, 1986.
17. M. Mohri. Finite-state transducers in language and speech processing. *Comp. Ling.*, 23(2):269–311, 1997.
18. M. Mohri. On the use of sequential transducers in natural language processing. In *Finite-State Language Processing*. MIT Press, 1997.
19. F. Pereira and M. Riley. Speech recognition by composition of weighted finite automata. In *Finite-State Language Processing*. MIT Press, 1997.
20. F. Pereira, M. Riley, and R. Sproat. Weighted rational transductions and their application to human language processing. In *Proc. ARPA HLT*, pages 249–54, 1994.
21. F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, 1988.
22. M. O. Rabin. Probabilistic automata. *Inf. and Control*, 6:230–45, 1963.
23. M. D. Riley, A. Ljolje, D. Hindle, and F. C. N. Pereira. The AT&T 60,000 word speech-to-text system. In *Proc. 4th EUROSPEECH*, volume 1, pages 207–210, 1995.
24. E. Roche. *Analyse Syntaxique Transformationelle du Francais par Transducteurs et Lexique-Grammaire*. PhD thesis, LITP-Université Paris 7, 1993.
25. A. Salomaa and M. Soittola. *Automata-Theoretic Aspects of Formal Power Series*. Springer-Verlag, 1978.
26. M. Silberztein. *Dictionnaires électroniques et analise automatique de textes: le systéme INTEX*. PhD thesis, Masson, Paris, France., 1993.
27. A. Weber and R. Klemm. Economy of description for single-valued transducers. *Inf. and Comp.*, 118:327–40, 1995.