p-Subsequentiable Transducers

Cyril Allauzen and Mehryar Mohri
{allauzen,mohri}@research.att.com

AT&T Labs – Research
180 Park Avenue
Florham Park, NJ 07932, USA

Abstract. p-subsequential transducers are efficient finite-state transducers with p final outputs used in a variety of applications. Not all transducers admit equivalent p-subsequential transducers however. We briefly describe an existing generalized determinization algorithm for p-subsequential transducers and give the first characterization of *p-subsequentiable transducers*, transducers that admit equivalent p-subsequential transducers. Our characterization shows the existence of an efficient algorithm for testing p-subsequentiability. We have fully implemented the generalized determinization algorithm and the algorithm for testing p-subsequentiability. We report experimental results showing that these algorithms are practical in large-vocabulary speech recognition applications. The theoretical formulation of our results is the equivalence of the following three properties for finite-state transducers: determinizability in the sense of the generalized algorithm, p-subsequentiability, and the twins property.

1 Introduction

Finite-state transducers are automata in which transitions are labeled with both an input and an output symbol. Transducers have been used successfully to create complex systems in many applications such as text and language processing, speech recognition and image processing [9, 8, 7, 12, 6].

The time efficiency of such systems is substantially increased when *subsequential transducers* [14], i.e. finite-state transducers with deterministic input, are used. Subsequential machines can be generalized to *p-subsequential transducers* which are transducers with deterministic input with p, $(p \geq 1)$, final output strings [10]. This generalization is necessary in many applications such as language processing to account for finite ambiguities [11].

Not all transducers admit equivalent p-subsequential transducers however. We present the first characterization of *p-subsequentiable transducers*, i.e. transducers that admit equivalent p-subsequential transducers. Our characterization is based on the twins property and leads to an efficient algorithm for testing p-subsequentiability. More generally, our results show the equivalence of the following three fundamental properties for finite-state transducers: determinizability in the sense of a generalized algorithm, p-subsequentiability, and the twins property.

This can also be viewed as a generalization of the results known in the case of functional transducers: determinizable functional transducers are exactly those that admit equivalent subsequential transducers [5]. We generalize these results by relaxing the condition on functionality: determinizable transducers are exactly those that admit equivalent p-subsequential transducers and exactly those that admit the twins property.

We have fully implemented the generalized determinization algorithm mentioned above and the algorithm for testing p-subsequentiability. We report experimental results showing that these algorithms are practical in large-vocabulary speech recognition applications.

We first introduce the notation used in the rest of this paper, then briefly describe a generalized determinization algorithm for p-subsequential transducers introduced by [10], present a fundamental characterization theorem, and describe our experimental results.

2 Preliminaries

Definition 1. *A* finite-state transducer $T = (\Sigma, \Delta, Q, I, F, E, \lambda, \rho)$ *is an 8-tuple where* Σ *is a finite input alphabet,* Δ *a finite output alphabet,* Q *a finite set of states,* $I \subseteq Q$ *the set of initial states,* $F \subseteq Q$ *the set of final states,* $E \subseteq Q \times \Sigma \times (\Delta \cup \{\epsilon\}) \times Q$ *a finite set of transitions,* $\lambda : I \to \Delta^*$ *the initial output function mapping* I *to* Δ^**, and* $\rho : F \to 2^{\Delta^*}$ *the final output function mapping each state* $q \in F$ *to a finite subset of* Δ^**.*

Given a transition $e \in E$, we denote by $i[e]$ its input label, $p[e]$ its origin or previous state and $n[e]$ its destination state or next state, $o[e]$ its output label. Given a state $q \in Q$, we denote by $E[q]$ the set of transitions leaving q. We extend the definitions of i, n, p, and E to sets in the following way: $i[\cup_{k \in K} e_k] = \cup_{k \in K} i[e_k]$ and similarly for n, p, and E.

A *path* $\pi = e_1 \cdots e_k$ in T is an element of E^* with consecutive transitions: $n[e_{i-1}] = p[e_i]$, $i = 2, \ldots, k$. We extend n and p to paths by setting: $n[\pi] = n[e_k]$ and $p[\pi] = p[e_1]$. We denote by $P(q, q')$ the set of paths from q to q' and by $P(q, x, q')$ the set of paths from q to q' with input label $x \in \Sigma^*$. These definitions can be extended to subsets $R, R' \subseteq Q$, by: $P(R, x, R') = \bigcup_{q \in R,\, q' \in R'} P(q, x, q')$. The labeling functions i and o can also be extended to paths by defining the label of a path as the concatenation of the labels of its constituent transitions:

$$i[\pi] = i[e_1] \cdots i[e_k] \qquad o[\pi] = o[e_1] \cdots o[e_k]$$

The set of output strings associated by a transducer T to an input string $x \in \Sigma^*$ is defined by:

$$[\![T]\!](x) = \bigcup_{\pi \in P(I,x,F)} \lambda(p[\pi]) \cdot o[\pi] \cdot \rho(n[\pi])$$

$[\![T]\!](x) = \emptyset$ when $P(I, x, F) = \emptyset$. The domain of definition of T is defined as: $Dom(T) = \{x \in \Sigma^* : [\![T]\!](x) \neq \emptyset\}$. A transducer is said to be *p-functional* for some integer p if it associates at most p strings to each input string, that is if $|[\![T]\!](x)| \leq p$ for any $x \in \Sigma^*$. Two transducers T and T' are *equivalent* when $[\![T]\!] = [\![T']\!]$.

A *successful path* in a transducer T is a path from an initial state to a final state. A state $a \in Q$ is *accessible* if q can be reached from I. It is *coaccessible* if a final state can be reached from q. T is *trim* if all the states of T are both accessible and coaccessible. T is *unambiguous* if for any string $x \in \Sigma^*$ there is at most one successful path labeled with x. An unambiguous transducer is thus p-functional,with $p = \max_{q \in F} |\rho(q)|$.

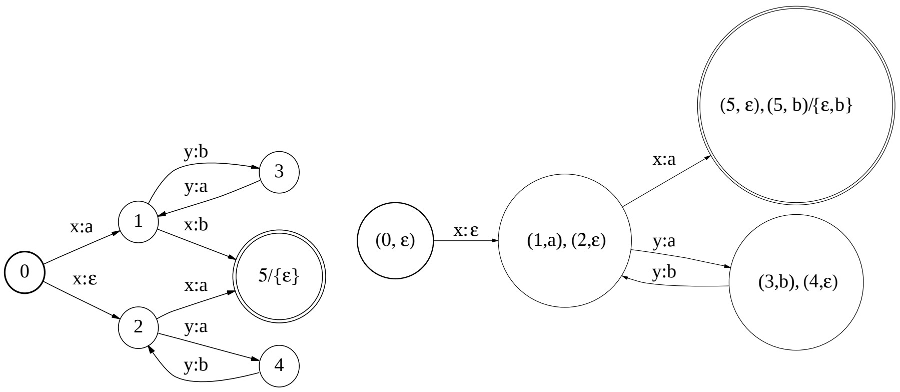
A transducer T is said to be *p-subsequential* [10] for some integer p if it has a unique initial state, no two transitions leaving the same state share the same input label and if there are at most p final output strings at each final state: $|\rho(f)| \leq p$ for all $f \in F$. T is said to be *p-subsequentiable* if there exists a p-subsequential transducer T' equivalent to T.

Given two strings x and y in Σ^*, we say that y is a *suffix* of x if there exists $z \in \Sigma^*$ such that $x = zy$ and similarly that y is a *prefix* of x if there exists z such that $x = yz$. We denote by $x \wedge y$ the longest common prefix of x and y and denote by $|x|$ the length of a string $x \in \Sigma^*$. We extend Σ by associating to each symbol $a \in \Sigma$ a new symbol denoted by a^{-1} and define Σ^{-1} as: $\Sigma^{-1} = \{a^{-1} : a \in \Sigma\}$. $X = (\Sigma \cup \Sigma^{-1})^*$ is then the set of strings written over the alphabet $(\Sigma \cup \Sigma^{-1})$. If we assume that $aa^{-1} = a^{-1}a = \epsilon$, then X forms a group called the *free group generated by* Σ and is denoted by $\Sigma^{(*)}$. Note that the inverse of a string $x = a_1 \cdots a_n$ is then $x^{-1} = a_n^{-1} \cdots a_1^{-1}$. The formula used in our definitions, theorems and proofs should be interpreted as equations in the free group generated by Σ^*.

3 General determinization algorithm with *p*-subsequential outputs

In this section, we give a brief description of a general determinization algorithm introduced by [10] that takes as input a transducer T and outputs a p-subsequential transducer $T' = (\Sigma, \Delta, Q', \{i'\}, F', E', \lambda', \rho')$. A transducer T for which the algorithm terminates and thus generates an equivalent p-subsequential transducer is said to be *determinizable*.

The algorithm is a generalization of the subset construction used in the determinization of finite automata. A state in the output transducer T' is a set of pairs (q, z) where q is a state of the input transducer T and $z \in \Sigma^*$ a remainder output string with the following property: if a state q' in T' containing a pair (q, z) can be reached from the initial state by a path with input x and output y, then q can be reached in T from an initial state by a path with input x and output yz. The following is the pseudocode of the algorithm.



(a) (b)

Fig. 1. Generalized determinization of finite-state transducers. (a) Non-deterministic transducer. (b) Construction of equivalent 2-subsequential transducer.

Transducer-Determinization(T)

1 $F' \leftarrow Q' \leftarrow E' \leftarrow \emptyset$
2 $S \leftarrow i' \leftarrow \{(i, \lambda(i)) : i \in I\}$
3 **while** $S \neq \emptyset$
4   **do** $p' \leftarrow head(S)$
5   Dequeue(S)
6   **for** each $x \in i[E[Q[p']]]$
7   **do** $y' \leftarrow \bigwedge\{z\,y : (p, z) \in p', (p, x, y, q) \in E\}$
8                 $q' \leftarrow \{(q, y'^{-1}\,z\,y) : (p, z) \in p', (p, x, y, q) \in E\}$
9                 $E' \leftarrow E' \cup \{(p', x, y', q')\}$
10   **if** $(q' \notin Q')$
11   **then** $Q' \leftarrow Q' \cup \{q'\}$
12   **if** $Q[q'] \cap F \neq \emptyset$
13   **then** $F' \leftarrow F' \cup \{q'\}$
14                                    $\rho'(q') \leftarrow \bigcup\{z\,\rho(q) : (q, z) \in q', q \in F\}$
15   Enqueue(S, q')
16 **return** T'

Line 1 initializes the set of states, final states, and transitions of T' to the empty set. The algorithm uses a queue S containing the set of states of T' to be considered next. S initially contains the unique initial state of T', i', which is the set of pairs of an initial state i of T and the corresponding initial output string $\lambda(i)$ (line 2).

Each time through the loop of lines 3-15, a new subset p' (or equivalently a new state of T') is extracted from S. The algorithm then creates (lines 6-9) a transition with input label $x \in \Sigma$ and output label $y' \in \Sigma^*$ leaving p' if there exists at least one pair $(p, z) \in p'$ such that p admits an outgoing transition with input label x and output label y. y' is then defined as the longest common prefix of all such zy's. The destination state q' of that transition is the subset containing the pairs $(q, y'^{-1}zy)$ such

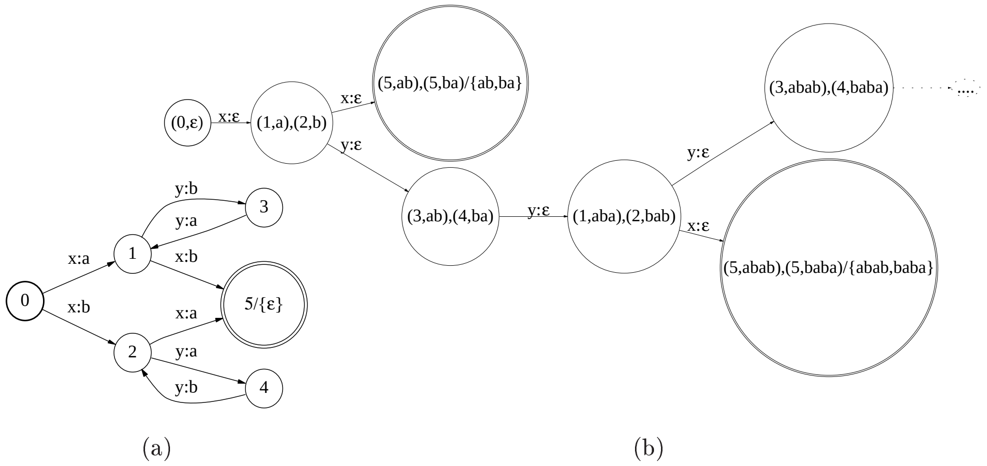


Fig. 2. Non-determinizable case. (a) A non-determinizable finite-state transducer; states 1 and 2 are non-twin siblings. (b) Determinization does not halt in this case and creates an infinite number of states.

that $(p, z) \in p'$ and (p, x, y, q) is a transition in E. If the destination state q' is new, it is added to Q' (lines 10-11). q' is a final state if it contains at least one pair (q, z), q being a final state. Its final set of output strings is then the union of $z\rho(q)$ over all such pairs (q, z).

There are input transducers that are not determinizable, that is for which the algorithm does not terminate. When it terminates, the output transducer T' is equivalent to T. Thus, it does not terminate with any transducer T that is not p-subsequentiable.

Figure 1 (b) illustrates the application of the algorithm to the transducer of figure 1 (a). Figures 2 (a)-(b) show an example of non-determinizable transducer.

The worst case complexity of determinization is exponential. However, in many applications such as large-vocabulary speech recognition such a blow-up does not occur and determinization leads to a significant improvement of speed versus accuracy at a reasonable cost in space [12].

4 Characterization

This section presents a characterization of p-subsequentiable transducers. The characterization is based on the following property.

Definition 2. *Let T be a finite-state transducer. Two states q_1 and q_2 of T are said to be* siblings *if there exist two strings x and y in Σ^* such that both q_1 and q_2 can be reached from I by paths with input label y and there are cycles at q_1 and q_2 both with input label x. Two siblings q_1 and q_2 are said to be* twins *if for any paths $\pi_1 \in P(I, y, q_1)$, $c_1 \in P(q_1, x, q_1)$, $\pi_2 \in P(I, y, q_2)$, $c_2 \in P(q_2, x, q_2)$,*

$$o[\pi_1]^{-1} o[\pi_2] = o[\pi_1 c_1]^{-1} o[\pi_2 c_2]$$

T has the twins property *if any two siblings in T are twins.*

The *twins property* was originally introduced by [4, 5] to give a characterization of functional subsequentiable transducers. The decidability of the twins property was also first proved by the same author (see also [3]). The first polynomial-time algorithm for testing the twins property was given by [15], this algorithm was later improved by [2]. More recently, we gave a significantly more efficient algorithm for testing the twins property based on the general algorithm of composition of finite-state transducers and a new characterization of the twins property in terms of combinatorics of words [1]

The following factorization lemma will be useful in several proofs.

Lemma 1. *Let $T = (\Sigma, \Delta, Q, I, F, E, \lambda, \rho)$ be a finite-state transducer, let π be a path from I to state $p \in Q$ and and π' a path from I to p' with the same input label $w = i[\pi] = i[\pi']$. Assume that $|w| > |Q|^2 - 1$, then there exist paths π_1, π_2, π_3, π'_1, π'_2, π'_3, such that:*

$$\pi = \pi_1\pi_2\pi_3 \quad \pi' = \pi'_1\pi'_2\pi'_3$$

where π_2 and π'_2 are cycles with non-empty input labels and: $i[\pi_k] = i[\pi'_k]$, for $k = 1, 2, 3$.

Proof. Consider the transducer U obtained by composing T and T^{-1}: $U = T \circ T^{-1}$. Since π and π' have the same input label, there exists a path ψ in U with input $o[\pi]$ and output $o[\pi']$. Since $|\psi| = |w| > |Q|^2 - 1$ and U has at most $|Q|^2$ states, ψ goes at least through one non-empty cycle ψ_2: $\psi = \psi_1\psi_2\psi_3$. This shows the existence of the common factoring for π and π' since ψ results from matching π and the path obtained from π' by swapping input and output labels. □

The following lemma will be used to prove that determinization terminates when the twins property holds.

Lemma 2. *Assume that T has the twins property. Let R be defined by:*

$$R = \{o[\pi']^{-1}o[\pi] : i[\pi] = i[\pi'] = w, |w| \leq |Q|^2\}$$

Let q_1 and q_2 be two states of T, π a path from I to q_1, and π' a path from I to q_2 with the same input label: $i[\pi] = i[\pi']$, then $o[\pi']^{-1}o[\pi] \in R$.

Proof. Let w be the common input label of π and π' and assume that $|w| > |Q|^2$. By lemma 1, paths π and π' can be factored in the following way:

$$\pi = \pi_1\pi_2\pi_3 \quad \pi' = \pi'_1\pi'_2\pi'_3$$

where π_2 and π'_2 are cycles with non-empty input labels and: $i[\pi_k] = i[\pi'_k]$, for $k = 1, 2, 3$. Let $\phi = \pi_1\pi_3 \in P(I, q_i)$ and $\phi' = \pi'_1\pi'_3 \in P(I, q_j)$ and $w' = i[\phi] = i[\phi']$. Since T has the twins property, $(o[\pi'_1\pi'_2])^{-1}o[\pi_1\pi_2] =$

$o[\pi_1']^{-1}o[\pi_1]$. Thus: $o[\pi_1'\pi_2'\pi_3']^{-1}o[\pi_1\pi_2\pi_3] = o[\pi_1'\pi_3']^{-1}o[\pi_1\pi_3] = o[\phi']^{-1}o[\phi]$. Since $|i[\pi_k]| > 0$, w' is a string strictly shorter than w. By induction, we can find paths $\phi \in P(I, q_1)$ and $\phi' \in P(I, q_2)$, with $i[\phi] = i[\phi'] = w'$, $|w'| \leq |Q|^2$ and such that $o[\pi']^{-1}o[\pi] = o[\phi']^{-1}o[\phi]$, thus $o[\pi']^{-1}o[\pi] \in R$. This proves the lemma. □

The following two lemmas are used in the proof of our main result.

Lemma 3. *Let $x_1, x_2, y_1, y_2 \in \Sigma^*$ be four strings over the alphabet Σ. Assume that for some integers $r \geq 0$ and $s > 0$, the following holds:*

$$(x_1 y_1^r)^{-1} x_2 y_2^r = (x_1 y_1^{r+s})^{-1} x_2 y_2^{r+s}$$

then:
$$x_1^{-1} x_2 = (x_1 y_1)^{-1} x_2 y_2.$$

Proof. Let $x_1, x_2, y_1, y_2 \in \Sigma^*$ be four strings satisfying the hypothesis of the lemma. We assume that $|x_2| \geq |x_1|$. The first equality can be rewritten as: $y_1^{-r} x_1^{-1} x_2 y_2^r = y_1^{-r-s} x_1^{-1} x_2 y_2^{r+s}$, or: $y_1^{r+s} y_1^{-r} x_1^{-1} x_2 = x_1^{-1} x_2 y_2^{r+s} y_2^{-r}$, and: $y_1^s (x_1^{-1} x_2) = (x_1^{-1} x_2) y_2^s$. This implies that $x_1^{-1} x_2$ is pure and, with the assumption that $|x_2| \geq |x_1|$, that $x_1^{-1} x_2 \in \Sigma^*$. Since the two members of the equality have the same length: $|y_1| = |y_2|$. Repeated applications of the equality lead to: $y_1^{sn}(x_1^{-1}x_2) = (x_1^{-1}x_2) y_2^{sn}$ for any $n \geq 0$. This shows that $x_1^{-1} x_2$ must be a prefix of a string in y_1^* and a suffix of a string in y_2^*, so there exist non-empty strings u and v with length less or equal to $|y_1| = |y_2|$ such that $x_1^{-1} x_2 u = y_1^p$ and $y_1 = vu$ for some integer p. It follows that $y_2 = uv$, which completes the proof of the lemma. □

Lemma 4. *Let $T' = (\Sigma, \Delta, Q', \{i'\}, F', E', \lambda', \rho')$ be a p-subsequential transducer equivalent to $T = (\Sigma, \Delta, Q, I, F, E, \lambda, \rho)$. Let $q \in F$ be a final state of T and $q' \in F'$ a final state of F' and assume that there exists $x \in \Sigma^*$ such that $P(I, x, q) \neq \emptyset$ in T and $P(i', x, q') \neq \emptyset$ in T'. Then, there exists $z \in \Delta^*$ such that for any paths $\pi \in P(I, q)$ and $\pi' \in P(i', q')$, if $i[\pi] = i[\pi']$ then $o[\pi]^{-1}o[\pi'] = z$.*

Proof. Let π and π' be two paths satisfying the hypothesis of the lemma. Since T and T' are equivalent, we have $o[\pi]\rho(q) \subseteq [\![T]\!](i[\pi]) = [\![T']\!](i[\pi])$. Since T' is p-subsequential we also have $[\![T']\!](i[\pi]) = o[\pi']\rho'(q')$. Hence, we have:

$$o[\pi]\rho(q) \subseteq o[\pi']\rho'(q')$$

and there exists $X \subseteq \rho'(q')$ such that $o[\pi]\rho(q) = o[\pi']X$. Let x and y be respectively the longest common prefixes of the strings in X and $\rho'(q')$, we then have $o[\pi]y = o[\pi']x$. Let $z = yx^{-1}$. Note that z only depends on q and q' and not on π and π'. We have $o[\pi]^{-1}o[\pi'] = z$. This proves the lemma. □

Our main characterization result of this section is given by the following theorem.

Theorem 1. *Let T be a trim finite-state transducer. Then the following three properties are equivalent:*

1. *T is determinizable;*
2. *T has the twins property;*
3. *T is p-subsequentiable.*

Proof. $1 \Rightarrow 3$: By definition of the algorithm, the output of determinization is a p-subsequential transducer.

$3 \Rightarrow 2$: Assume that T is p-subsequentiable, then there exists a p-subsequential transducer T' equivalent to T. Let q_1 and q_2 be two siblings in T. Consider four paths π_1, c_1, π_2, c_2 as in the definition of the twins property. Since T is trim, there exist a path π'_1 from q_1 to a final state and a path π'_2 from q_2 to a final state. Since T' is equivalent to T, there must be a path π in T from the initial state to a state q with input label xy^r with $r \geq 0$, a cycle c at q with input label y^s with $s > 0$, and two paths μ_1 and μ_2 from q to a final state with input labels respectively $i[\pi'_1]$ and $i[\pi'_2]$. We then have:

$$i[\pi_1 c_1^r \pi'_1] = i[\pi\mu_1] \quad \text{and} \quad i[\pi_1 c_1^{r+s} \pi'_1] = i[\pi c \mu_1]$$

and we can apply lemma 4 with the final states $n[\pi'_1]$ and $n[\mu_1]$. Thus, there exists $z_1 \in \Delta^*$ such that:

$$z_1 = o[\pi_1 c_1^r \pi'_1]^{-1} o[\pi\mu_1] = o[\pi_1 c_1^{r+s} \pi'_1]^{-1} o[\pi c \mu_1]$$

from which we infer that:

$$o[\pi'_1] z_1 o[\mu_1]^{-1} = o[\pi_1 c_1^r]^{-1} o[\pi] = o[\pi_1 c_1^{r+s}]^{-1} o[\pi c]$$

We can prove in a similar way that:

$$o[\pi_2 c_2^r]^{-1} o[\pi] = o[\pi_2 c_2^{r+s}]^{-1} o[\pi c]$$

Thus:

$$o[\pi_1 c_1^{r+s}]^{-1} o[\pi_2 c_2^{r+s}] = o[\pi_1 c_1^r]^{-1} o[\pi_2 c_2^r]$$

By lemma 3:

$$o[\pi_1]^{-1} o[\pi_2] = o[\pi_1 c_1]^{-1} o[\pi_2 c_2]$$

Thus, any two siblings q_1 and q_2 in T are twins, and T has the twins property.

$2 \Rightarrow 1$: Assume that T has the twins property. Let $\{(q_1, z_1), \ldots, (q_n, z_n)\}$ be a subset created during the execution of the determinization algorithm. By construction, states $q_1, \ldots, q_n$ can all be reached from I by paths labeled with the same input string w.[1] Let z be defined by:

$$z = \bigwedge_{p[\pi] \in I, i[\pi] = w} o[\pi]$$

[1] Note that we may have $q_i = q_j$ for some choices of i and j.

By definition of the algorithm, for $i = 1, \ldots, n$ there exists a path π_i from I to q_i with input label w such that:

$$z_i = z^{-1} o[\pi_i]$$

Let $\Pi = \pi_i$ and $\Pi' = \pi_j$ for some $i, j = 1, \ldots, n$, we have:

$$z_j^{-1} z_i = o[\pi']^{-1} o[\pi]$$

Thus, by lemma 2, $z_j^{-1} z_i \in R$ with:

$$R = \{o[\pi']^{-1} o[\pi] : i[\pi] = i[\pi'] = w, |w| \leq |Q|^2\}$$

Define K as the maximum length of the elements of R: $K = \max_{x \in R} |x|$. Since the remainders in the same subset cannot have a common non-empty prefix, for any remainder string z_i, there exists at least on remainder z_j such that $z_i \wedge z_j = \epsilon$. Since $z_j^{-1} z_i \in R$, $|z_j^{-1} z_i| = |z_j| + |z_i| \leq K$. Thus, for $i = 1, \ldots, n$, $|z_i| \leq K$, any remainder in a subset belongs to $\Sigma^{\leq K}$, and a subset necessarily belongs to $2^{|Q| \times \Sigma^{\leq K}}$, which is a finite set. This guarantees the termination of the algorithm and thus the determinizability of T. □

5 Experiments and results

We have fully implemented the general determinization algorithm presented in section 3. We used a priority queue implemented with a heap to sort the transitions leaving each subset and another priority queue to sort the final output strings. Since the computation of the transitions leaving a subset only depends on the states and remainder strings of that subset and on the input transducer, one can limit the computation of the result to just the part that is needed. Thus, we gave an on-the-fly implementation of the algorithm which was incorporated in the FSM library [13].

Our experiments in large-vocabulary speech recognition showed the algorithm to be quite efficient. It took about 5s using a Pentium III 700MHz with 2048 Kb of cache and 4Gb of RAM to construct a p-subsequential transducer equivalent to a transducer T with 440,000 transitions representing the mapping from phonemic sequences to word sequences obtained by composition of two transducers.

We also implemented an efficient algorithm for testing the twins property [1]. With our implementation, the p-subsequentiality of the transducer T already described could be tested in just 60s using the same machine.

6 Conclusion

A new characterization of p-subsequentiable transducers was given. The twins property was shown to be a necessary and sufficient condition for the p-subsequentiability of a finite-state transducer without requiring it to be p-functional and a necessary and sufficient condition for the general determinizability of transducers. We reported experimental results demonstrating the practicality of our algorithms for testing p-subsequentiability and for determinizing transducers in large-vocabulary speech recognition applications.

References

1. Cyril Allauzen and Mehryar Mohri. On the Determinizability of Weighted Automata and Transducers. In *Proceedings of the workshop Weighted Automata: Theory and Applications (WATA)*, Dresden, Germany, March 2002.
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. Christian Choffrut. Une caractérisation des fonctions séquentielles et des fonctions sous-séquentielles en tant que relations rationnelles. *Theoretical Computer Science*, 5:325–338, 1977.
5. 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.
6. 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.
7. Maurice Gross and Dominique Perrin, editors. *Electronic Dictionnaries and Automata in Computational Linguistics*, volume 377 of *Lecture Notes in Computer Science*. Springer Verlag, 1989.
8. Ronald M. Kaplan and Martin Kay. Regular Models of Phonological Rule Systems. *Computational Linguistics*, 20(3):331–378, 1994.
9. Lauri Karttunen. The Replace Operator. In *33rd Annual Meeting of the Association for Computational Linguistics*, pages 16–23. Association for Computational Linguistics, 1995. Distributed by Morgan Kaufmann Publishers, San Francisco, California.
10. Mehryar Mohri. On some Applications of Finite-State Automata Theory to Natural Language Processing. *Journal of Natural Language Engineering*, 2:1–20, 1996.
11. Mehryar Mohri. Finite-State Transducers in Language and Speech Processing. *Computational Linguistics*, 23(2), 1997.
12. Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley. Weighted Finite-State Transducers in Speech Recognition. *Computer Speech and Language*, 16(1):69–88, 2002.
13. Mohri, Mehryar and Fernando C. N. Pereira and Michael Riley. General-Purpose Finite-State Machine Software Tools. *http://www.research.att.com/sw/tools/fsm*, AT&T Labs – Research, 1997.
14. Marcel Paul Schützenberger. Sur une variante des fonctions séquentielles. *Theoretical Computer Science*, 4(1):47–57, 1977.
15. Andreas Weber and Reinhard Klemm. Economy of Description for Single-Valued Transducers. *Information and Computation*, 118(2):327–340, 1995.