# A Uniformization Theorem for Nested Word to Word Transductions

Dmitry Chistikov and Rupak Majumdar

Max Planck Institute for Software Systems (MPI-SWS)
Kaiserslautern and Saarbrücken, Germany
{dch,rupak}@mpi-sws.org

**Abstract.** We study the class of relations implemented by nested word to word transducers (also known as visibly pushdown transducers). We show that any such relation can be uniformized by a functional relation from the same class, implemented by an unambiguous transducer. We give an exponential upper bound on the state complexity of the uniformization, improving a previous doubly exponential upper bound. Our construction generalizes a classical construction by Schützenberger for the disambiguation of nondeterministic finite-state automata, using determinization and summarization constructions on nested word automata. Besides theoretical interest, our procedure can be the basis for synthesis procedures for nested word to word transductions.

**Keywords:** uniformization, transduction, nested word, visibly pushdown language.

## 1 Introduction

A central result in the theory of rational languages is the *unambiguity theorem* [1,2], which states that every rational function can be implemented by an unambiguous transducer, that is, a transducer that has at most one successful run on any input. Schützenberger [3] gave an elegant and direct proof of the unambiguity theorem, by showing, for any rational function, a matrix representation which can be made unambiguous. Sakarovitch [2] subsequently showed that the construction of Schützenberger can be used as the foundation for several results in the theory of rational functions, most notably the rational uniformization theorem and the rational cross-section theorem (see [4,5,2]).

In more detail, the construction of Schützenberger starts with a (possibly nondeterministic) finite-state transducer $\mathcal{T}$, and constructs a new transducer with the property that any input string has at most one successful run. The construction performs a cross product of the subset automaton with the original automaton, and shows that certain edges in the cross product can be removed to obtain unambiguity, while preserving the language. As a simple consequence, the input–output relation that relates two words $(u, w)$ if the transducer $\mathcal{T}$ can output $w$ on input $u$, can be *uniformized*: for every $u$ in the domain of $\mathcal{T}$, we can pick a unique $w$ that is related to it (the rational *uniformization theorem*).

In this paper, we present an extension of the construction by Schützenberger to transducers from *nested words* to words. A nested word consists of a word over an alphabet, together with a *matching relation* that relates "call" positions in the word with corresponding "return" positions. Nested word automata [6] were introduced as a subclass of pushdown automata which are expressive enough for a large number of application domains but nevertheless retain many closure and decidability properties of regular languages. Nested word automata distinguish between "call," "return," and "internal" positions of the word, and, informally, push a symbol on the runtime stack on a call letter, pop a symbol on a return letter, and do not touch the stack on an internal letter.

These automata were extended to nested word *transducers* by Raskin and Servais [7] (under the name "visibly pushdown transducers"). The definition of nested word transducers in [7] allows transitions with $\varepsilon$-marking, but ensures that no transition can read and write symbols of different types (call, return, and internal symbols, in terms of visibly pushdown automata). We study a model of nested-word to word transducers considered in [8,9], in which the output word carries no structural information (i. e., call, return, or internal). That is, our transducers transform nested words into "normal" (linear) words.

Our main construction generalizes Schützenberger's construction to give an unambiguous automaton that is at most a single exponential larger than the input automaton. Our construction relies on the standard determinization construction for nested word automata from [6], as well as on a *summarization* of a nested word automaton, which captures the available properly-nested computation paths between two states. We show how to prune the product of these automata, in analogy with [3,2], to get an unambiguous automaton.

As a consequence of our construction, we obtain a uniformization theorem for nested-word to word transducers: any relation defined by such a transducer can be uniformized by a functional (i. e., single-valued) relation implemented by an unambiguous transducer. For functional nested-word to word transductions this yields an unambiguity theorem: every single-valued transduction can be implemented by a unambiguous transducer. The increase in the number of states is at most exponential, which improves a doubly exponential construction of [10] based on a notion of look-ahead. Note that for several other classes of algebraic (i. e., pushdown) transductions, uniformization theorems were obtained in [11].

In this extended abstract we focus on the case of nested words without unmatched calls, which we call *closed* words. Our construction for this special case captures main technical ideas and can be extended with an auxiliary transformation to fit the general case. Another variant of our construction, also not discussed here, applies to so-called weakly hierarchical nested word automata from [6,12], which are restricted to record only the current state on the stack.

Besides theoretical interest, our results provide an easily-implementable disambiguation construction. Since uniformization results are at the core of reactive synthesis techniques [13], our construction can form the basis of implementations of synthesis procedures for nested-word to word transductions.

The structure of the current paper is as follows. After introducing necessary definitions, we describe three basic constructions on nested word automata in Section 3. We show how to combine them to obtain a specific disambiguation of an arbitrary automaton (the *Schützenberger construction*) in Section 4. The proofs are delayed until Section 6, while in Section 5 we discuss how to use the construction to obtain uniformization and unambiguity theorems. Short notes sketching the extension to the general (non-closed-word) case can be found in relevant parts of Sections 4 and 5.

## 2 Nested Words and Transducers

A *nested word* of length $k$ over an alphabet $\Sigma$ is a pair $u = (x, \nu)$, where $x \in \Sigma^k$ and $\nu$ is a *matching relation* of length $k$, that is, a subset $\nu \subseteq \{-\infty, 1, \ldots, k\} \times \{1, \ldots, k, +\infty\}$ such that, first, if $\nu(i, j)$ holds, then $i < j$; second, for $1 \leq i \leq k$ each of the sets $\{j \mid \nu(i, j)\}$ and $\{j \mid \nu(j, i)\}$ contains at most one element; third, whenever $\nu(i, j)$ and $\nu(i', j')$, it cannot be the case that $i < i' \leq j < j'$. We assume that $\nu(-\infty, +\infty)$ never holds.

If $\nu(i, j)$, then the position $i$ in the word $u$ is said to be a *call*, and the position $j$ a *return*. All positions from $\{1, \ldots, k\}$ that are neither calls nor returns are *internal*. A call (a return) is *matched* if $\nu$ matches it to an element of $\{1, \ldots, k\}$ and *unmatched* otherwise.

We shall call a nested word *closed* if it has no unmatched calls, and *well-matched* if it has no unmatched calls and no unmatched returns. We denote the set of all nested words over $\Sigma$ by $\Sigma^{*\mathsf{n}}$, the set of all closed nested words by $\Sigma^{*\mathsf{c}}$, and the set of all well-matched words by $\Sigma^{*\mathsf{w}}$. Observe that $\Sigma^{*\mathsf{w}} \subsetneq \Sigma^{*\mathsf{c}} \subsetneq \Sigma^{*\mathsf{n}}$.

The family of all non-empty (word) languages over the alphabet $\Delta$ is denoted by $\mathcal{L}(\Delta^*)$.

Define a (*nested-word to word*) *transducer* over the input alphabet $\Sigma$ and output alphabet $\Delta$ as a structure $\mathcal{T} = (Q, P, \delta, Q^{\mathsf{i}}, Q^{\mathsf{f}}, P^{\mathsf{i}})$, where:

- $Q$ is a finite non-empty set of (linear) states,
- $P$ is a finite set of hierarchical states,
- $\delta = (\delta^{\mathsf{call}}, \delta^{\mathsf{int}}, \delta^{\mathsf{ret}})$, where
  - $\delta^{\mathsf{int}} \subseteq Q \times \Sigma \times Q \times \mathcal{L}(\Delta^*)$ is a set of internal transitions,
  - $\delta^{\mathsf{call}} \subseteq Q \times \Sigma \times Q \times P \times \mathcal{L}(\Delta^*)$ is a set of call transitions,
  - $\delta^{\mathsf{ret}} \subseteq P \times Q \times \Sigma \times Q \times \mathcal{L}(\Delta^*)$ is a set of return transitions,
- $Q^{\mathsf{i}} \subseteq Q$ and $Q^{\mathsf{f}} \subseteq Q$ are sets of initial and final linear states, and
- $P^{\mathsf{i}} \subseteq P$ is a set of initial hierarchical states.

A *path* through a transducer $\mathcal{T}$ driven by an input word $u = (a_1 \ldots a_k, \nu) \in \Sigma^{*\mathsf{n}}$ is a sequence of alternating linear states and transitions of $\mathcal{T}$, where $i$th transition leads from the current linear state to the next one, carries the letter $a_i \in \Sigma$, and has type chosen according to the matching relation $\nu$; furthermore, for every pair of matched call and return, hierarchical states encountered in the corresponding transitions are required to be the same (we say that they are *sent* and *received* along the hierarchical edges).

The path is *successful* if it starts in a state from $Q^{\mathsf{i}}$, ends in a state from $Q^{\mathsf{f}}$, and all states received along the hierarchical edges in unmatched returns belong to $P^{\mathsf{i}}$. Note that, following [6], we impose no requirement on unmatched calls (our automata are called *linearly accepting*). A (*successful*) *computation* consists of a (successful) path and a sequence of words $w_i$ taken from languages $\ell_i \in \mathcal{L}(\Delta^*)$ in the transitions of the path in the same order. The concatenation of these words gives a word $w \in \Delta^*$, which is said to be *output* by the transducer.

We say that the transducer $\mathcal{T}$ *implements* the transduction $T \subseteq \Sigma^{*\mathsf{n}} \times \Delta^*$ that contains each pair $(u, w)$ if and only if there exists a successful computation of $\mathcal{T}$ driven by the input $u$ and having the output $w$. When we are only interested in the behaviour of $\mathcal{T}$ on closed words, we say that $\mathcal{T}$ *weakly implements* $T^{\mathsf{c}} = T \cap (\Sigma^{*\mathsf{c}} \times \Delta^*)$.

## 3 Automata and Auxiliary Constructions

A *nested word automaton* (NWA, or simply an *automaton*) $\mathcal{A}$ is defined similarly to a transducer, with the only difference that it has no output, that is, all $\mathcal{L}(\Delta^*)$ factors are dropped. Words $u \in \Sigma^{*\mathsf{n}}$ carried by successful paths are said to be *accepted* by $\mathcal{A}$, and the automaton itself is then said to *recognize* the language $L \subseteq \Sigma^{*\mathsf{n}}$ of all such words. Two automata are *equivalent* if they recognize the same language.

We call $\mathcal{A}$ (*weakly*) *unambiguous* if it has at most one successful path for each (closed) word $u \in \Sigma^{*\mathsf{n}}$ ($u \in \Sigma^{*\mathsf{c}}$). As usual, $\mathcal{A}$ is *deterministic* if, first, each of the sets $Q^{\mathsf{i}}$ and $P^{\mathsf{i}}$ contains at most one element and, second, for every $q \in Q$, $a \in \Sigma$, and $p \in P$, each of the three sets $\delta^{\mathsf{int}}$, $\delta^{\mathsf{call}}$, and $\delta^{\mathsf{ret}}$ contains at most one tuple of the form $(q, a, q')$, $(q, a, q', p')$, and $(p, q, a, q')$, respectively. Every deterministic automaton is unambiguous, but not vice versa. Also recall that a linear state $q$ of an automaton $\mathcal{A}$ is called *accessible* if there exists a path in $\mathcal{A}$ starting in some linear state $q_0 \in Q^{\mathsf{i}}$ and ending in $q$, in which all states received along the hierarchical edges in unmatched returns belong to $P^{\mathsf{i}}$.

We now define three auxiliary constructions for nested word automata. Their combination, as shown in Sections 4 and 5, can be used to obtain uniformization and unambiguity theorems for nested-word to word transductions.

**Determinization.** Every nested word automaton is known to be equivalent to a deterministic one, by a variant of the well-known subset construction [6],[1] which can be traced to a 1985 paper [14]. Given an automaton $\mathcal{A} = (Q, P, \delta, Q^{\mathsf{i}}, Q^{\mathsf{f}}, P^{\mathsf{i}})$, construct the automaton with the following components:

- the set of linear states is $2^{Q \times Q}$ (in this context pairs $(q, q') \in Q \times Q$ are called *summaries* and understood as available properly-nested path fragments between pairs of states);
- the set of hierarchical states is $\{p'_0\} \cup (2^{Q \times Q} \times \Sigma)$, where the state $p'_0$ is new;

[1] Note that the construction described in the print version of [6] is flawed, so we refer the reader to the electronic document available on the Web.

- for every input letter $a$ and every state $S \subseteq Q \times Q$, transitions lead from $S$ to states $S'$ defined as follows:
  - for an internal transition, $S'$ contains all summaries $(q, q'')$ such that there exists a summary $(q, q') \in S$ and an internal transition $(q', a, q'') \in \delta^{\mathsf{int}}$;
  - for a call transition, $S'$ contains all summaries $(q'', q'')$, for which there exists a summary $(q, q') \in S$ and a call transition $(q', a, q'', p) \in \delta^{\mathsf{call}}$; along the hierarchical edge the pair $(S, a)$ is sent;
  - for a return transition upon the receipt of a hierarchical state $H = (S^0, b)$, where $S^0 \subseteq Q \times Q$, the state $S'$ contains all summaries $(q_0, q'')$ such that there exist summaries $(q_0, q_1) \in S^0$ and $(q, q') \in S$, a call transition $(q_1, b, q, p) \in \delta^{\mathsf{call}}$ and a return transition $(p, q', a, q'') \in \delta^{\mathsf{ret}}$ with a matching $p \in P$;
  - for a return transition upon the receipt of a hierarchical state $p'_0$, the state $S'$ contains all summaries $(q, q'')$ such that there exists a summary $(q, q') \in S$ and a return transition $(p_0, q', a, q'') \in \delta^{\mathsf{ret}}$ with some $p_0 \in P^{\mathsf{i}}$;
- the only initial linear state is $\{(q_0, q_0) \mid q_0 \in Q^{\mathsf{i}}\}$ (here we deviate from [6], where the set $Q^{\mathsf{i}} \times Q^{\mathsf{i}}$ is used), and an arbitrary linear state $S$ is final whenever it contains some pair $(q, q')$ with $q' \in Q^{\mathsf{f}}$;
- the only initial hierarchical state is $p'_0$.

The accessible part of this automaton is called the *determinization* of $\mathcal{A}$ and denoted $\mathcal{A}_{\mathsf{det}}$. It is deterministic and can be proved equivalent to $\mathcal{A}$.

**Summarization.** We also introduce another auxiliary automaton, closely related to that of $\mathcal{A}_{\mathsf{det}}$. This automaton will keep track of summaries instead of single states, similarly to $\mathcal{A}_{\mathsf{det}}$, but still rely on nondeterminism rather than subset construction to mimic the behaviour of $\mathcal{A}$. In short, for any transition in $\mathcal{A}_{\mathsf{det}}$, if it leads from a state containing a summary $(q, q')$ to some state $S'$, this automaton will have transitions from $(q, q')$ to all summaries that can use $(q, q')$ as a witness for their inclusion in $S'$. More formally, this summary-tracking automaton has the following components:

- the set of linear states is $Q \times Q$;
- the set of hierarchical states is $(\{p'_0\} \times \delta^{\mathsf{ret}}) \cup ((Q \times Q) \times \delta^{\mathsf{call}})$, where $p'_0$ is new;
- for every input letter $a$ and every state $(q, q') \in Q \times Q$, transitions from $(q, q')$ are defined as follows:
  - every internal transition $(q', a, q'') \in \delta^{\mathsf{int}}$ of $\mathcal{A}$ is translated here into an internal transition leading to $(q, q'')$;
  - every call transition $t = (q', a, q'', p) \in \delta^{\mathsf{call}}$ is translated into a call transition leading to $(q'', q'')$, with $((q, q'), t)$ sent along the hierarchical edge;
  - every pair of call and return transitions $t = (q_1, b, q, p) \in \delta^{\mathsf{call}}$ and $(p, q', a, q'') \in \delta^{\mathsf{ret}}$ with matching $p \in P$ is translated, for all $q_0 \in Q$, into return transitions to $(q_0, q'')$, depending on the state $((q_0, q_1), t)$ received along the hierarchical edge;

  - every return transition $t = (p_0, q', a, q'') \in \delta^{\mathsf{ret}}$ with $p_0 \in P^{\mathsf{i}}$ is also translated into a return transition to $(q, q'')$ with $(p'_0, t)$ received along the hierarchical edge;
- initial linear states are $(q_0, q_0)$, for all $q_0 \in Q^{\mathsf{i}}$, and the set of final linear states is $Q \times Q^{\mathsf{f}}$;
- the set of initial hierarchical states is $\{p'_0\} \times \delta^{\mathsf{ret}}$.

The accessible part of this automaton is called the *summarization* of $\mathcal{A}$ and denoted $\mathcal{A}_{\mathsf{sum}}$. This automaton is also easily shown to be equivalent to $\mathcal{A}$.

Note that we could also define $\mathcal{A}_{\mathsf{sum}}$ in a more natural way, with the set of hierarchical states $\{p'_0\} \cup ((Q \times Q) \times \Sigma)$, similarly to $\mathcal{A}_{\mathsf{det}}$. However, we need to tie transitions of $\mathcal{A}_{\mathsf{sum}}$ to individual transitions of $\mathcal{A}$, as seen in the proof of Lemma 7 in Section 6 below.

**Product.** Let us define the product of two nested word automata $\mathcal{A}_1 = (Q_1, P_1, \delta_1, Q^{\mathsf{i}}_1, Q^{\mathsf{f}}_1, P^{\mathsf{i}}_1)$ and $\mathcal{A}_2 = (Q_2, P_2, \delta_2, Q^{\mathsf{i}}_2, Q^{\mathsf{f}}_2, P^{\mathsf{i}}_2)$ in the standard way:

- sets of linear and hierarchical states are $Q_1 \times Q_2$ and $P_1 \times P_2$, respectively;
- whenever $\delta_1$ and $\delta_2$ contain transitions from $q_1$ to $q'_1$ and from $q_2$ to $q'_2$, respectively, having the same type and carrying the same letter $a \in \Sigma$, this results in a transition from $(q_1, q_2)$ to $(q'_1, q'_2)$ of the same type carrying $a \in \Sigma$; the state sent or received along the hierarchical edge (if any) is the ordered pair of the hierarchical states carried by these transitions;
- sets of initial and final linear states are $Q^{\mathsf{i}}_1 \times Q^{\mathsf{i}}_2$ and $Q^{\mathsf{f}}_1 \times Q^{\mathsf{f}}_2$;
- the set of initial hierarchical states is $P^{\mathsf{i}}_1 \times P^{\mathsf{i}}_2$.

The accessible part of this automaton is called the *product* of $\mathcal{A}_1$ and $\mathcal{A}_2$ and denoted $\mathcal{A}_1 \times \mathcal{A}_2$.

## 4 The Schützenberger Construction for NWA

We now describe a special disambiguation construction for nested word automata, generalizing a construction due to Schützenberger [3,2].

First suppose that $\mathcal{B}$ and $\mathcal{A}$ are nested word automata, and there exists a mapping $\mu$ that takes states and transitions of $\mathcal{B}$ to states and transitions of $\mathcal{A}$, preserving letters and transition types, in such a way that the image of a successful path is always a successful path. Then we say that the automaton $\mathcal{B}$ is *morphed* into $\mathcal{A}$ by $\mu$ (this word usage roughly corresponds to that in [2,5]). We shall sometimes decorate symbols denoting attributes of $\mathcal{B}$ with the prime $'$, to distinguish these attributes from those of $\mathcal{A}$.

We now define a special operation eliminating some transitions of an automaton. Suppose that $\mathcal{B}$ is morphed into $\mathcal{A}$ by $\mu$, and $\delta'$ is some subset of $\mathcal{B}$'s transitions. For every state $s'$ of $\mathcal{B}$, consider its image $\mu(s')$ in $\mathcal{A}$ and take an arbitrary transition $t$ arriving at $\mu(s')$. Denote the set of all transitions arriving at $s'$ by $\textsc{in}(s')$. If the set $\mu^{-1}(t) \cap \textsc{in}(s') \cap \delta'$ has cardinality 2 or greater, i. e., if the inverse image $\mu^{-1}(t)$ contains more than one transition from $\delta'$ arriving at $s'$, then transform the automaton $\mathcal{B}$ by eliminating all of these transitions but one,

arbitrarily chosen. Repeat this procedure for all states $s'$ of $\mathcal{B}$ and all transitions $t$ arriving at their images $\mu(s')$. We shall say that the resulting automaton is obtained from $\mathcal{B}$ by *weeding* transitions $\delta'$ with respect to the mapping $\mu$.

Now everything is ready for our generalization of the Schützenberger construction to nested word automata. Suppose we are given an automaton $\mathcal{A}$. First construct the automaton $\mathcal{A}_{\mathsf{sum}} \times \mathcal{A}_{\mathsf{det}}$, where $\mathcal{A}_{\mathsf{sum}}$ and $\mathcal{A}_{\mathsf{det}}$ are the summarization and determinization of $\mathcal{A}$. Let the mapping $\pi$ take linear states and transitions of $\mathcal{A}_{\mathsf{sum}} \times \mathcal{A}_{\mathsf{det}}$ to their projections in $\mathcal{A}_{\mathsf{det}}$. Weed the sets of internal and return transitions of $\mathcal{A}_{\mathsf{sum}} \times \mathcal{A}_{\mathsf{det}}$ with respect to $\pi$ (note that all call transitions are left intact), and then for every final state $S$ of $\mathcal{A}_{\mathsf{det}}$, make all but one state in $\pi^{-1}(S)$ non-final (this remaining final state should have the form $((q, q'), S)$, where $(q, q')$ is final in $\mathcal{A}_{\mathsf{sum}}$). Denote the obtained automaton by $\mathcal{S}$.

**Theorem 1.** *The automaton $\mathcal{S}$ is weakly unambiguous, equivalent to and morphed into $\mathcal{A}$.*

The proof of this theorem is given in Section 6. Before that, in Section 5, we show that $\mathcal{S}$ can be used to obtain a uniformization of any relation weakly implemented by a transducer whose underlying automaton is $\mathcal{A}$. (Recall that this relation is defined as a subset of $\Sigma^{*\mathsf{c}} \times \Delta^*$.)

Note that if we cannot disregard non-closed words, then an auxiliary step is needed. Roughly speaking, if in $\mathcal{S}$ we additionally separate matched calls from unmatched calls and weed the latter in the same fashion as earlier, then the obtained automaton will be unambiguous, equivalent to and morphed into $\mathcal{A}$. We leave a detailed discussion of this construction until the full version of the paper, and only note that this transformation involves at most a constant-factor increase in the number of states and transitions.

**Proof idea.** The "morphism" part is easy, and the main challenge is the weak unambiguity and equivalence to $\mathcal{A}$. We borrow the overall strategy from [2,5], but prefer not to hide the needed properties behind the framework of covering of automata, and instead make the main steps explicit to achieve more clarity.

Consider the mapping $\pi$ defined above. For every final state $S$ of $\mathcal{A}_{\mathsf{det}}$, exactly one of the states in its inverse image $\pi^{-1}(S)$ is final. Now for every successful path in $\mathcal{A}_{\mathsf{det}}$ terminating in $S$ we construct its inverse image in $\mathcal{S}$. If we show that at all states every arriving transition in $\mathcal{A}_{\mathsf{det}}$ has a non-empty inverse image in $\mathcal{A}_{\mathsf{sum}} \times \mathcal{A}_{\mathsf{det}}$ (this is almost *in-surjectivity* in terms of [2,5]), then our definition of weeding will ensure that at each state exactly one option remains in $\mathcal{S}$ (*in-bijectivity*). As a result, every successful path in $\mathcal{A}_{\mathsf{det}}$ has exactly one counterpart in $\mathcal{S}$. Since the automaton $\mathcal{A}_{\mathsf{sum}} \times \mathcal{A}_{\mathsf{det}}$ only accepts words accepted by $\mathcal{A}$, the equivalence follows; and the unambiguity then follows from that of $\mathcal{A}_{\mathsf{det}}$.

There are, however, two issues with this strategy. Note that the concatenation of correct path segments is not necessarily a correct path segment, unlike in the standard finite-state machines. To this end, we choose not to weed call transitions, in order to ensure that the path can be correctly prolonged on each step, with matching hierarchical states sent and received along hierarchical edges. This explains why our construction of $\mathcal{S}$ guarantees only weak unambiguity.

At this point we also brush upon the more subtle second issue, which is the reason for our having defined and taken $\mathcal{A}_{\mathsf{sum}}$ in the product, as opposed to just using $\mathcal{A}$ as in the original construction. Imagine we did otherwise, and consider some return transition from $(q', S')$ to $(q'', S'')$ in $\mathcal{S}$ encountered while constructing the inverse image of some path in $\mathcal{A}_{\mathsf{det}}$. Note that the specific transition from $q'$ to $q''$ is in fact chosen at the weeding stage and fixes some specific state $p \in P$ received along the hierarchical edge.

Now if $S'$ contains two summaries $(q_1, q')$ and $(q_2, q')$, we cannot know which of $q_1$ and $q_2$ we will hit after extending the path backwards until the matching call. But then it may well be the case that none of the call transitions arriving in this state $q_i$ under the appropriate input letter carries the hierarchical state $p$. This means that the weeding of return transitions would change the recognized language, which is undesirable. Introducing the automaton $\mathcal{A}_{\mathsf{sum}}$, which keeps track of summaries, resolves this issue, as seen from Lemmas 5 and 6 in Section 6.

## 5 Uniformization of Transductions

Return to the problem of uniformizing an arbitrary nested-word to word transduction $T$. Recall that a relation $U \subseteq A \times B$ is a *uniformization* of a relation $T \subseteq A \times B$ if $U$ is a subset of $T$, single-valued as a transduction, and has the same domain, that is, if $U \subseteq T$ and for every $u \in A$, the existence of a $w \in B$ such that $(u, w) \in T$ implies that there is exactly one $w' \in B$ such that $(u, w') \in U$.

In this section, we show how to use the Schützenberger construction to obtain a uniformization of an arbitrary nested-word to word transduction $T$. Take any transducer $\mathcal{T}$ implementing $T$, and denote by $\mathcal{A}$ its underlying automaton, that is, one obtained by removing the output labels from transitions. Transform $\mathcal{A}$ into $\mathcal{S}$ as described in Section 4. By the "morphism" part of Theorem 1, $\mathcal{S}$ is morphed into $\mathcal{A}$ by some mapping $\rho$, so that each transition $t'$ in $\mathcal{S}$ is projected onto a single transition $\rho(t')$ in $\mathcal{A}$. Take the output label $\ell \subseteq \Delta^*$ of $\rho(t')$ in $\mathcal{T}$ and specify any single word $w \in \ell$ as the output of $t'$. The automaton $\mathcal{S}$ is thus transformed into a transducer $\mathcal{U}$, which is claimed to satisfy our needs.

**Theorem 2.** *Let the transducer $\mathcal{T}$ weakly implement a transduction $T$. Then the transducer $\mathcal{U}$ weakly implements a transduction $U$, which is a uniformization of $T$.*

*Proof.* We use the "equivalence" and "weak unambiguity" parts of Theorem 1. First observe that any nested word $u$ belongs to the domain of $T$ (or, respectively, $U$) if and only if it is is accepted by $\mathcal{A}$ (or, respectively, $\mathcal{S}$). It then follows from the "equivalence" part that $T$ and $U$ have the same domain within $\Sigma^{*\mathsf{n}}$. Second, if $(u, w')$ and $(u, w'')$ with $w' \neq w''$ are in $U$, then $\mathcal{U}$ has at least two successful paths driven by $u$. This is impossible for any closed word $u$ by the "weak unambiguity" part of Theorem 1. □

We say that a relation $U \subseteq \Sigma^{*\mathsf{c}} \times \Delta^*$ is a *weak uniformization* of a transduction $T \subseteq \Sigma^{*\mathsf{n}} \times \Delta^*$ if $U$ is a uniformization of the transduction $T^{\mathsf{c}} = T \cap (\Sigma^{*\mathsf{c}} \times \Delta^*)$.

**Corollary.** *Any nested-word to word transduction $T$ implemented by a transducer with $n$ linear states has a weak uniformization implemented by a transducer with at most $n^2\,2^{n^2-1}$ linear and $n\,2^{n^2-1}\,|\delta^{\mathsf{call}}| + |\delta^{\mathsf{ret}}|$ hierarchical states.*

We conclude this section with a series of remarks. First, the statement of Theorem 1 can actually be augmented with the following observation. The transduction $U$ implemented by $\mathcal{U}$ not only has the property that $U^{\mathsf{c}}$ is a uniformization of $T^{\mathsf{c}}$, but also satisfies $U \subseteq T$ and has the same domain as $T$. In other words, for all non-closed words $u$, the existence of a $w \in \Delta^*$ such that $(u, w) \in T$ implies that there is at least one $w' \in \Delta^*$ such that $(u, w') \in U$, and all such pairs $(u, w')$ from $U$ also belong to $T$.

Second, an additional refinement of the construction of $\mathcal{S}$, as sketched in Section 4, can be used to obtain a stronger version of Theorem 2, giving a uniformization instead of a weak uniformization. The proof of the stronger version repeats the proof above almost literally.

Finally, note that for a single-valued transduction $T$, the statement above gives an unambiguity theorem: the construction defines a transducer implementing $T$, whose underlying automaton is (weakly) unambiguous.

## 6 Proof of Theorem 1

In this section we demonstrate that the automaton $\mathcal{S}$ constructed in Section 4 has the desired properties: weak unambiguity, equivalence to the original automaton $\mathcal{A}$, and the property of being morphed into $\mathcal{A}$. We first prove an important property of accessible states in the automaton $\mathcal{A}_{\mathsf{sum}} \times \mathcal{A}_{\mathsf{det}}$.

**Lemma 1.** *If a state $((q, q'), S)$ is accessible in $\mathcal{A}_{\mathsf{sum}} \times \mathcal{A}_{\mathsf{det}}$, then $(q, q')$ belongs to $S$ and $S$ is accessible in $\mathcal{A}_{\mathsf{det}}$.*

*Proof.* It is clear that no state of the form $((q, q'), S)$ can be accessible in $\mathcal{A}_{\mathsf{sum}} \times \mathcal{A}_{\mathsf{det}}$ unless the state $S$ is accessible in $\mathcal{A}_{\mathsf{det}}$. The fact that $(q, q') \in S$ is then proved by induction on the length of the path leading from the initial state of $\mathcal{A}_{\mathsf{det}}$ to $S$. Indeed, $\{(q_0, q_0) \mid q_0 \in Q^{\mathsf{i}}\}$ is the only initial state of $\mathcal{A}_{\mathsf{det}}$ and the states $(q_0, q_0)$, $q_0 \in Q^{\mathsf{i}}$, are initial in $\mathcal{A}_{\mathsf{sum}}$. This forms the induction base. Further, suppose that $((q_1, q_2), S)$ is accessible and a transition leads from this state to some state $((q, q'), S')$. Then $(q_1, q_2) \in S$ and, by the definition of $\mathcal{A}_{\mathsf{sum}} \times \mathcal{A}_{\mathsf{det}}$, there are transitions from $S$ to $S'$ and from $(q_1, q_2)$ to $(q, q')$. It remains to use the fact that whenever a transition leads in $\mathcal{A}_{\mathsf{sum}}$ from $(q_1, q_2)$ to $(q, q')$, any corresponding transition in $\mathcal{A}_{\mathsf{det}}$ from a state containing $(q_1, q_2)$ can only lead to a state containing $(q, q')$. □

Lemma 1 shows that every accessible state of $\mathcal{A}_{\mathsf{sum}} \times \mathcal{A}_{\mathsf{det}}$ can be regarded as a set of summaries $S$ with a distinguished element $(q, q') \in S$. From now on we shall only use the states of $\mathcal{A}_{\mathsf{sum}} \times \mathcal{A}_{\mathsf{det}}$ satisfying the conclusion of this lemma. For our second lemma, recall that by $\pi$ we denote the mapping taking every state of $\mathcal{A}_{\mathsf{sum}} \times \mathcal{A}_{\mathsf{det}}$ to its projection in $\mathcal{A}_{\mathsf{det}}$.

**Lemma 2 (in-surjectivity).** *For every accessible state $((q, q'), S)$ of $\mathcal{A}_{\mathsf{sum}} \times \mathcal{A}_{\mathsf{det}}$ and every transition $t$ arriving at its projection $S$ in $\mathcal{A}_{\mathsf{det}}$, there exists at least one transition in $\mathcal{A}_{\mathsf{sum}} \times \mathcal{A}_{\mathsf{det}}$ whose projection is $t$.*

*Proof.* First apply Lemma 1 to note that $(q, q') \in S$. Let the transition $t$ arriving at $S$ depart from a state $S^0$. We claim that there exists a summary $(q_1, q_2) \in S^0$ and a transition from $(q_1, q_2)$ to $(q, q')$ in $\mathcal{A}_{\mathsf{sum}}$ such that the inverse image $\pi^{-1}(t)$ in $\mathcal{A}_{\mathsf{sum}} \times \mathcal{A}_{\mathsf{det}}$ contains a transition from $((q_1, q_2), S^0)$ to $((q, q'), S)$.

Indeed, since $(q, q') \in S$, it follows that there exists a summary $(q_1, q_2) \in S^0$ that guaranteed the inclusion of $(q, q')$ into $S$ by the definition of $\mathcal{A}_{\mathsf{det}}$. The definition of $\mathcal{A}_{\mathsf{sum}}$ then ensures the existence of a transition of the same type from this $(q_1, q_2)$ to $(q, q')$. The rest follows by the definition of $\mathcal{A}_{\mathsf{sum}} \times \mathcal{A}_{\mathsf{det}}$. □

**Lemma 3 (partial in-bijectivity).** *For every state $((q, q'), S)$ of $\mathcal{S}$ and every internal or return transition $t$ arriving at its projection $S$ in $\mathcal{A}_{\mathsf{det}}$, there exists a unique transition in $\mathcal{S}$ whose projection is $t$.*

*Proof.* Follows from Lemma 2 and the definitions of $\mathcal{S}$ and weeding. □

**Lemma 4.** *The automaton $\mathcal{S}$ is weakly unambiguous.*

*Proof.* Consider any closed word $u \in \Sigma^{*\mathsf{c}}$ accepted by $\mathcal{S}$ and assume for the sake of contradiction that there are two different successful paths driven by $u$. Observe that the projection by means of $\pi$ of a successful path in $\mathcal{S}$ is a successful path in $\mathcal{A}_{\mathsf{det}}$. Since the automaton $\mathcal{A}_{\mathsf{det}}$ is deterministic and, therefore, unambiguous, it follows that both paths in $\mathcal{S}$ are projected onto a single path in $\mathcal{A}_{\mathsf{det}}$.

Consider the last position in this pair of paths, on which these paths disagree, that is, $((q_1, q_1'), S) \neq ((q_2, q_2'), S)$ (note that the projection onto $\mathcal{A}_{\mathsf{det}}$ should be the same, hence the common component $S$). The successors of these two states are the same in both paths, and the following segments up to the end of the paths also coincide (both paths end in the same final state of $\mathcal{S}$, since for each $S$ at most one state from $\pi^{-1}(S)$ is final). Suppose that our chosen pair of paths leads from this pair of states to some state $((q, q'), S')$.

Now consider the transition $t$ leading from $S$ to $S'$ in the (single) induced path in $\mathcal{A}_{\mathsf{det}}$. This transition has at least two elements in its inverse image $\pi^{-1}(t)$, one of them departing from $((q_1, q_1'), S)$ and another from $((q_2, q_2'), S)$. By the partial in-bijectivity given by Lemma 3, this transition $t$ can only be a (matched) call transition. We shall show, however, that this conclusion leads to a contradiction.

Indeed, consider the tails of the paths, which coincide by our initial assumptions. It follows that these tails take the same transition at the return position matching the call in question (recall that our word is closed, so all calls are matched). This transition receives some state $((q_0, q_0'), t_0, S^0, a)$ along the hierarchical edge, and so the call in question can only originate at the state $((q_0, q_0'), S^0)$. This contradicts the availability of the choice between $((q_1, q_1'), S)$ and $((q_2, q_2'), S)$ and establishes that one of the paths must be invalid. □

Our next goal is to show that every successful path in $\mathcal{A}_{\mathsf{det}}$ has an inverse image in $\mathcal{S}$ that is also a successful path. To prove this fact, we need an auxiliary lemma that describes the behaviour of $\mathcal{A}_{\mathsf{sum}}$ on well-matched words.

**Lemma 5.** *Suppose that the automaton $\mathcal{A}_{\mathsf{sum}}$ can be driven by some well-matched word $u \in \Sigma^{*\mathsf{w}}$ from a state $(q_0, q_1)$ to a state $(q_2, q_3)$. Then $q_0 = q_2$.*

*Proof.* Any well-matched word is a concatenation of internal transitions and well-matched fragments enclosed in matched pairs of symbols. By the definition of $\mathcal{A}_{\mathsf{sum}}$, internal transitions do not change the first component of a summary, and every matched return resets it to the state just before the matching call. □

**Lemma 6.** *The automaton $\mathcal{S}$ is equivalent to $\mathcal{A}$.*

*Proof.* Since the projection of a successful path through $\mathcal{S}$ is a successful path through $\mathcal{A}_{\mathsf{det}}$, it is sufficient to demonstrate that $\mathcal{S}$ always accepts whenever $\mathcal{A}_{\mathsf{det}}$ accepts. Consider a successful path through $\mathcal{A}_{\mathsf{det}}$ and construct a path through $\mathcal{S}$ as follows. First consider the destination of the path in $\mathcal{A}_{\mathsf{det}}$ and take the (unique) final state in its inverse image. Second, reconstruct a path through $\mathcal{S}$ by the following procedure. On each step, given a state of $\mathcal{S}$ and a transition $t$ arriving at its projection in $\mathcal{A}_{\mathsf{det}}$, choose a transition from the inverse image $\pi^{-1}(t)$ to obtain the previous state in the path through $\mathcal{S}$. Lemmas 2 and 3 reveal that this procedure will indeed yield a path-like sequence in $\mathcal{S}$. Since all summaries from the initial state of $\mathcal{A}_{\mathsf{det}}$ are initial states of $\mathcal{A}_{\mathsf{sum}}$, this sequence will begin in an initial state of $\mathcal{S}$. However, we also need to show that this sequence will indeed be a correct path through $\mathcal{S}$, that is, states sent and received along the hierarchical edges will match.

Consider a fragment of the input word of the form $\langle bua\rangle$, where $u \in \Sigma^{*\mathsf{w}}$ (here and below angle brackets decorate call and return positions). Suppose that the automaton $\mathcal{A}_{\mathsf{det}}$ is driven by $a\rangle$ from a state $S$ to a state $S'$, and the reconstructed path in $\mathcal{S}$ distinguishes summaries $(q, q') \in S$ and $(q_0, q'') \in S'$, with $\mathcal{A}_{\mathsf{sum}}$ driven by $a\rangle$ from the former to the latter. Also suppose that at this point a transition receiving a state $((q_0, q_1), t_0, S^0, b)$ along the hierarchical edge is chosen in $\mathcal{S}$. Now assume that the matching call drives $\mathcal{A}_{\mathsf{det}}$ from $S^0$ to some state $S''$. We need to show that when the reconstruction of the path in $\mathcal{S}$ reaches this call, the state $S''$ will have been mapped to the state $((q, q), S'')$ and a transition to this state from $((q_0, q_1), S^0)$ will be available, with $((q_0, q_1), t_0, S^0, b)$ sent along the hierarchical edge.

The second of this claims is relatively straightforward. Indeed, by the definitions of $\mathcal{A}_{\mathsf{det}}$ and $\mathcal{A}_{\mathsf{sum}}$, a transition from $((q, q'), S)$ to $((q_0, q''), S')$ driven by $a\rangle$ upon the receipt of $((q_0, q_1), t_0, S^0, b)$ along the hierarchical edge is witnessed by a pair of call and return transitions of the form $t_0 = (q_1, b, q, p) \in \delta^{\mathsf{call}}$, $(p, q', a, q'') \in \delta^{\mathsf{ret}}$ with matching $p \in P$. Since a call transition from $S^0$ to $S''$ is available in $\mathcal{A}_{\mathsf{det}}$ with an input letter $b$, it follows that a call transition with the same input letter leads from $((q_0, q_1), S^0)$ to $((q, q), S'')$ in $\mathcal{S}$.

Now turn to the first of the claims. Recall that we are now dealing with a fragment of the input word of the form $\langle bua\rangle$ with $u \in \Sigma^{*\mathsf{w}}$. Observe that our entire argument can be interpreted as a proof that the reconstructed sequence segment induces a correct path segment in $\mathcal{A}_{\mathsf{sum}}$. We now use induction to prove this fact. The base case corresponds to words containing internal symbols only, and does not require any analysis. The inductive step corresponds to words of

the form $\langle bua\rangle$ with $u \in \Sigma^{*\mathsf{w}}$, as specified earlier. Here, since just before the input symbol $a\rangle$ the automaton $\mathcal{A}_{\mathsf{sum}}$ is set to the state $(q, q')$ and the word $u$ is well-matched, one concludes with the help of the inductive hypothesis and Lemma 5 that after reading the symbol $\langle b$ the automaton $\mathcal{A}_{\mathsf{sum}}$ must have been in some state of the form $(q, \bar{q})$, where $\bar{q} \in Q$. Since this state $(q, \bar{q})$ is taken from $S''$, and the state $S''$ is the destination of some call transition in $\mathcal{A}_{\mathsf{det}}$, it follows from the definition of $\mathcal{A}_{\mathsf{det}}$ that $\bar{q} = q$. This concludes the proof of Lemma 6. □

**Lemma 7.** *The automaton $\mathcal{S}$ is morphed into $\mathcal{A}$.*

*Proof.* Removing $\mathcal{A}_{\mathsf{det}}$-components maps the states of $\mathcal{S}$ into the set of states of $\mathcal{A}_{\mathsf{sum}}$. Transitions are mapped to transitions of $\mathcal{A}_{\mathsf{sum}}$ accordingly. It remains to observe that $\mathcal{A}_{\mathsf{sum}}$ is itself morphed into $\mathcal{A}$, for every transition of $\mathcal{A}_{\mathsf{sum}}$ can be mapped into a specific transition of $\mathcal{A}$. □

Theorem 1 follows from Lemmas 4, 6 and 7.

## References


1. Kobayashi, K.: Classification of formal languages by functional binary transducers. Information and Control 15, 95–109 (1969)
2. Sakarovitch, J.: A construction on finite automata that has remained hidden. Theoretical Computer Science 204, 205–231 (1998)
3. Schützenberger, M.P.: Sur les relations rationnelles entre monoïdes libres. Theoretical Computer Science 3, 243–259 (1976)
4. Eilenberg, S.: Automata, Languages, and Machines. Vol. A. Academic Press (1974)
5. Sakarovitch, J.: Elements of Automata Theory. Cambridge University Press (2009)
6. Alur, R., Madhusudan, P.: Adding nesting structure to words. Journal of the ACM 56(3), 16 (2009); revised version available online at `http://robotics.upenn.edu/~alur/Jacm09.pdf`
7. Raskin, J.-F., Servais, F.: Visibly pushdown transducers. In: L. Aceto et al. (Eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 386–397. Springer, Heidelberg (2008)
8. Staworko, S., Laurence, G., Lemay, A., Niehren, J.: Equivalence of deterministic nested word to word transducers. In: Kutyłowski, M., Gębala, M., Charatonik, W. (Eds.) FCT 2009. LNCS, vol. 5699, pp. 310–322. Springer, Heidelberg (2009)
9. Filiot, E., Raskin, J.-F., Reynier, P.-A., Servais, F., Talbot, J.-M.: Properties of visibly pushdown transducers. In: Hliněný, P., Kučera, A. (Eds.) MFCS 2010. LNCS, vol. 6281, pp. 355–367. Springer, Heidelberg (2010)
10. Filiot, E., Servais, F.: Visibly pushdown transducers with look-ahead. In: M. Bieliková et al. (Eds.) SOFSEM 2012. LNCS, vol. 7147, pp. 251–263. Springer, Heidelberg (2012)
11. Konstantinidis, S., Santean, N., Yu, S.: Representation and uniformization of algebraic transductions. Acta Informatica 43, 395–417 (2007)
12. Alur, R., Madhusudan, P.: Adding nesting structure to words. In: Ibarra, O.H., Dang, Z. (Eds.) DLT 2006. LNCS, vol. 4036, pp. 1–13. Springer, Heidelberg (2006)
13. Thomas, W.: Facets of synthesis: revisiting Church's problem. In: L. de Alfaro (Ed.) FOSSACS 2009. LNCS, vol. 5504, pp. 1–14. Springer, Heidelberg (2009)
14. Von Braunmühl, B., Verbeek, R.: Input driven languages are recognized in $\log n$ space. Annals of Discrete Mathematics 24, 1–20 (1985)

## Appendix

This Appendix is devoted to the general case, as opposed to the case of closed nested words considered in the main part of this extended abstract.

**Separation.** Let us begin with the following definition. We say that a nested word automaton *separates unmatched returns* if none of its call transitions send an initial hierarchical state along the hierarchical edge. In such an automaton sets on transitions invoked on unmatched returns and matched returns do not intersect, for they carry hierarchical states from two disjoint sets, $P^{\mathsf{i}}$ and $P \setminus P^{\mathsf{i}}$. Note that for an arbitrary $\mathcal{A}$, its determinization $\mathcal{A}_{\mathsf{det}}$ separates unmatched returns.

We shall also need a somewhat dual property, that of separating unmatched calls. To this end, we define the following transformation, which we call the separation of a nested word automaton. In the result of separation sets of transitions that can be (successfully) invoked on matched and unmatched calls do not intersect.

Assume that an automaton $\mathcal{A} = (Q, P, \delta, Q^{\mathsf{i}}, Q^{\mathsf{f}}, P^{\mathsf{i}})$ is known to separate unmatched returns in the sense defined above. Take $\mathcal{A}$ and construct the new automaton as follows:

- every linear state $q \in Q$ is *annotated* with symbols from $\{\mathsf{main}, \mathsf{sub}\}$, i. e., split into two, $q_{\mathsf{main}}$ and $q_{\mathsf{sub}}$;
- every hierarchical state $p \in P$ is annotated with symbols from $\{\mathsf{z}, \mathsf{n}, \mathsf{uc}, \mathsf{ur}\}$;
- transitions are replaced in the following way:
  - $(q, a, q') \in \delta^{\mathsf{int}}$ with $(q_{\mathsf{main}}, a, q'_{\mathsf{main}})$ and $(q_{\mathsf{sub}}, a, q'_{\mathsf{sub}})$;
  - $(q, a, q', p) \in \delta^{\mathsf{call}}$ with $(q_{\mathsf{main}}, a, q'_{\mathsf{main}}, p_{\mathsf{uc}})$, $(q_{\mathsf{main}}, a, q'_{\mathsf{sub}}, p_{\mathsf{z}})$, and $(q_{\mathsf{sub}}, a, q'_{\mathsf{sub}}, p_{\mathsf{n}})$;
  - $(p, q, a, q') \in \delta^{\mathsf{ret}}$, $p \notin P^{\mathsf{i}}$, with $(p_{\mathsf{n}}, q_{\mathsf{sub}}, a, q'_{\mathsf{sub}})$ and $(p_{\mathsf{z}}, q_{\mathsf{sub}}, a, q'_{\mathsf{main}})$;
  - $(p, q, a, q') \in \delta^{\mathsf{ret}}$, $p \in P^{\mathsf{i}}$, with $(p_{\mathsf{ur}}, q_{\mathsf{main}}, a, q'_{\mathsf{main}})$;
- sets of initial and final states are $\{q_{\mathsf{main}} \mid q \in Q^{\mathsf{i}}\}$ and $\{q_{\mathsf{main}} \mid q \in Q^{\mathsf{f}}\}$, respectively;
- the set of initial hierarchical states is $\{p_{\mathsf{ur}} \mid p \in P^{\mathsf{i}}\}$.

We call the obtained automaton the *separation* of $\mathcal{A}$ and denote it by $\mathcal{A}_{\mathsf{sep}}$.

*Claim.* Sets of successful paths through $\mathcal{A}_{\mathsf{sep}}$ and $\mathcal{A}$ are set in bijection by the mapping $\tau$ stripping annotations from states of $\mathcal{A}_{\mathsf{sep}}$. Furthermore, for every $u \in \Sigma^{*\mathsf{n}}$ accepted by $\mathcal{A}$ and every successful path through $\mathcal{A}_{\mathsf{sep}}$ driven by $u$, all unmatched calls, unmatched returns, and matched calls and returns carry states from disjoint sets $P_{\mathsf{uc}}$, $P_{\mathsf{ur}}$, and $P_{\mathsf{z}} \cup P_{\mathsf{n}}$ on the hierarchical edges, respectively.

**The Schützenberger construction.** Now we are ready to describe the full version of our modification of the Schützenberger construction. Given an arbitrary automaton $\mathcal{A}$, construct the automaton $\mathcal{S}$ as described in Section 4. Take the separation $(\mathcal{S})_{\mathsf{sep}}$ and consider the set $\delta^{\mathsf{uc}}$ of all call transitions sending a state of the form $p_{\mathsf{uc}}$ along the hierarchical edge. Define the mapping $\pi_0$ on the

linear states of $(\mathcal{S})_{\mathsf{sep}}$ by the rule $\pi_0\big(((q,q'),S)_{\mathsf{main}}\big) = \pi_0\big(((q,q'),S)_{\mathsf{sub}}\big) = S$ and extend it to transitions in the straightforward fashion. Now weed the set of transitions $\delta^{\mathsf{uc}}$ with respect to the mapping $\pi_0$ and denote the obtained automaton by $\mathcal{S}'$.

**Theorem 3.** *The automaton $\mathcal{S}'$ is unambiguous, equivalent to and morphed into $\mathcal{A}$.*

The proof of this theorem is given below.

**Uniformization of transductions.** Note that substituting $\mathcal{S}'$ for $\mathcal{S}$ in the transformations described in Section 5, one can transform an arbitrary nested-word to word transducer $\mathcal{T}$ into a transducer $\mathcal{U}'$, which is defined in analogy to $\mathcal{U}$. Using Theorem 3 one can then obtain the following theorem.

**Theorem 4.** *Let the transducer $\mathcal{T}$ implement a transduction $T$. Then the transducer $\mathcal{U}'$ implements a transduction $U$, which is a uniformization of $T$.*

**Corollary.** *Any nested-word to word transduction $T$ implemented by a transducer with $n$ linear states has a uniformization implemented by a transducer with at most $n^2\,2^{n^2}$ linear and $3\,n\,2^{n^2-1}\,|\delta^{\mathsf{call}}| + |\delta^{\mathsf{ret}}|$ hierarchical states.*

As hinted in Section 5, the proof of Theorem 4 repeats the proof of Theorem 2 almost literally. Remarks given at the end of Section 5 also apply to Theorem 4.

**Proof of Theorem 3.** The rest of Appendix is devoted to the proof of Theorem 3. We split the proof into three lemmas, one for each of the properties of $\mathcal{S}'$.

For the first lemma we need the mapping $\pi_0$ defined earlier in Section 4. This mapping strips annotations from states of $\mathcal{S}'$ and then removes $\mathcal{A}_{\mathsf{sum}}$-components of states, thus sending states of $\mathcal{S}'$ to states of $\mathcal{A}_{\mathsf{det}}$.

**Lemma 8.** *The automaton $\mathcal{S}'$ is unambiguous.*

*Proof.* It follows from Lemma 4 and the properties of the separation construction that $\mathcal{S}'$ is at least weakly unambiguous. Now take any non-closed word $u \in \Sigma^{*\mathsf{n}}$ and consider any two paths through $\mathcal{S}'$ driven by $u$. By the properties of the separation construction, these two paths have different counterparts in $\mathcal{S}$. However, their projections onto $\mathcal{A}_{\mathsf{det}}$ by means of $\pi_0$ are the same, since $\mathcal{A}_{\mathsf{det}}$ is deterministic and, therefore, unambiguous.

Consider the last position, on which these two paths disagree. Transitions in these paths corresponding to this position are different, but projected to the same transition in $\mathcal{A}_{\mathsf{det}}$. As shown in the proof of Lemma 4, this projection cannot be an internal or a return transition, nor a matched call. It can only be an unmatched call, but in this case two different unmatched call transitions are sent by $\pi_0$ to a single transition of $\mathcal{A}_{\mathsf{det}}$. This contradicts our definition of $\mathcal{S}'$, since one of these transitions must have been weeded out. □

**Lemma 9.** *The automaton $\mathcal{S}'$ is equivalent to $\mathcal{A}$.*

*Proof.* Recall that by Lemma 6 the automaton $\mathcal{S}$ is equivalent to $\mathcal{A}$, and the automaton $\mathcal{S}'$ is obtained by eliminating some transitions from $(\mathcal{S})_{\mathsf{sep}}$. It is then sufficient to demonstrate that $\mathcal{S}'$ accepts whenever $\mathcal{A}$ accepts or, equivalently, whenever $\mathcal{A}_{\mathsf{det}}$ accepts.

Consider any successful path through $\mathcal{A}_{\mathsf{det}}$ and construct a corresponding path through $\mathcal{S}'$, similarly to the proof of Lemma 6. Note that here annotations of states are chosen unambiguously, according to the matching structure of the input word. For matched call, internal, and return transitions, the existence of an appropriate transition in $\mathcal{S}'$ follows from the previous arguments. Among unmatched calls, some of the transitions in $\mathcal{S}'$ may have been weeded out, but at least one option on each step is still available. □

**Lemma 10.** *The automaton $\mathcal{S}'$ is morphed into $\mathcal{A}$.*

*Proof.* Stripping annotations from the states of $\mathcal{S}'$ turns them into states of the automaton $\mathcal{S}$, which is morphed into $\mathcal{A}$ by Lemma 7. □

Theorem 3 follows from Lemmas 8, 9 and 10.