Visibly Pushdown Transducers with Well-nested Outputs

Pierre-Alain Reynier * and Jean-Marc Talbot *

Aix Marseille Université, CNRS, LIF UMR 7279, 13288, Marseille, France

Abstract. Visibly pushdown transducers (VPTs) are visibly pushdown automata extended with outputs. They have been introduced to model transformations of nested words, i.e. words with a call/return structure. When outputs are also structured and well nested words, VPTs are a natural formalism to express tree transformations evaluated in streaming. We prove the class of VPTs with well-nested outputs to be decidable in PTIME. Moreover, we show that this class is closed under composition and that its type-checking against visibly pushdown languages is decidable.

1 Introduction

Visibly pushdown automata (VPA) [1] (first introduced as input-driven pushdown automata [3]) are pushdown machines whose stack behavior is synchronized with the structure of the input word. More precisely, the input alphabet is partitioned into call and return symbols; when reading a call symbol the machine must push a symbol onto the stack, when reading a return symbol it must pop a symbol from the stack and when reading an internal symbol the stack remains unchanged. Such words over a structure alphabet are called nested words.

Visibly pushdown transducers (VPTs) [7,9,6,10] extend visibly pushdown automata with outputs. Each transition is equipped with an output word; a VPT thus transforms an input word into an output word obtained as the concatenation of all the output words produced along a successful run (*i.e.* a sequence of transitions) on this input. VPTs are a strict subclass of pushdown transducers (PTs) and strictly extend finite state transducers. Several problems that are undecidable for PTs are decidable for VPTs similarly to finite state transducers: functionality (in PTIME), k-valuedness (in co-NPTIME) and functional equivalence (EXPTIME-complete) [6]. However, some decidability results or valuable properties of finite-state transducers do not hold for VPTs [7]: VPTs are not closed under composition and type-checking against VPA is undecidable (deciding whether the range of a transducer is included into the language of a given VPA).

Unranked trees and more generally hedges can be linearized into nested words over a structured alphabet (such as XML documents). These words for which the matching between call and return symbols is perfect are called well-nested words. So, VPTs are a suitable formalism to express hedge transformations. Moreover, as they process the linearization from left to right, they are also an adequate formalism to model and analyze transformations in streaming, as shown in [5]. VPTs output strings; operating on well-nested inputs, they define hedge-to-string transformations. If the output strings are well-nested too, they define hedge-to-hedge transformations [4].

* Partly supported by the PEPS project ”Synthesis of Stream Processors” funded by CNRS.

In [6], by means of a syntactical restriction on transition rules, a class of VPTs whose range contains only well-nested words is presented. This class enjoys good properties: it is closed under composition and type-checking against visibly pushdown languages is decidable. One may then wonder whether these properties come from this particular subclass or from the fact that the range of these VPTs contains only well-nested words.

In this paper, we consider two classes of transductions (that is, of relations) over nested words definable by VPTs. First, the class of *globally well-nested* transductions, denoted $\mathcal{G}_{\mathsf{wn}}$, is the class of VPT transductions whose range contains only well-nested words. The second class, named *almost well-nested* and denoted $\mathcal{A}_{\mathsf{wn}}$, slightly generalizes the first one as follows: there must exist $k \in \mathbb{N}$ such that every output word contains at most k unmatched returns and at most k unmatched calls. These two classes of transductions naturally define two classes of transducers gwnVPT and awnVPT: a VPT is in gwnVPT (resp. in awnVPT) if the transduction it represents is in $\mathcal{G}_{\mathsf{wn}}$ (resp. in $\mathcal{A}_{\mathsf{wn}}$). While defined in a semantical way, we provide criteria on successful computations of VPTs characterizing precisely the classes gwnVPT and awnVPT. Then, based on these criteria, we prove the class awnVPT to be decidable in PSPACE. Regarding the class gwnVPT, using a recent result of [2], we prove it is decidable in PTIME. Finally, we prove that the two classes gwnVPT and awnVPT enjoy good properties: they are closed under composition and type-checking is decidable against visibly pushdown languages.

The paper is organized as follows: definitions and recalls of some basic properties on VPTs are presented in Section 2. We introduce in Section 3 the two classes of transductions we define in this paper as well as the corresponding classes of transducers. Considering additionally the (restricted) class introduced in [6], we prove also that they form a strict hierarchy. Then, we give in Section 4 a precise characterization of the classes gwnVPT and awnVPT by means of some criteria on VPTs. Section 5 describes decision procedure of the considered classes of transducers. Finally, the closure of the considered classes under composition and the decidability of type-checking are addressed in Section 6. Omitted details can be found in a technical report [8].

2 Preliminaries

(Well) nested words The set of all finite words (resp. of all words of length at most n) over a finite alphabet Σ is denoted by Σ^* (resp. $\Sigma^{\leq n}$); the empty word is denoted by ϵ. A *structured alphabet* is a triple $\Sigma = (\Sigma_c, \Sigma_i, \Sigma_r)$ of disjoint alphabets, of call, internal and return symbols respectively. Given a structured alphabet Σ, we always denote by Σ_c, Σ_i and Σ_r its implicit structure, and identify Σ with $\Sigma_c \cup \Sigma_i \cup \Sigma_r$. A *nested word* is a finite word over a structured alphabet.

The set of *well-nested words* over a structured alphabet Σ is the least set, denoted by Σ^*_{wn}, that satisfies (i) $\epsilon \in \Sigma^*_{\mathsf{wn}}$, (ii) for all $i \in \Sigma_i$, $w \in \Sigma^*_{\mathsf{wn}}$, $iw \in \Sigma^*_{\mathsf{wn}}$, and (iii) for all $w, w' \in \Sigma^*_{\mathsf{wn}}$, $c \in \Sigma_c$, $r \in \Sigma_r$, $cwrw' \in \Sigma^*_{\mathsf{wn}}$. E.g. on $\Sigma = (\{c_1, c_2\}, \emptyset, \{r\})$, the nested word $c_1 r c_2 r$ is well-nested while $r c_1$ is not.

For a word w from Σ^*, we define its balance B as the difference between the number of symbols from Σ_c and of symbols from Σ_r occurring in w. Note that if $w \in \Sigma^*_{\mathsf{wn}}$, then $\mathsf{B}(w) = 0$; but the converse is false as exemplified by $r c_1$.

Lemma 1. *Let $u, v \in \Sigma^*$. We have* $\mathsf{B}(uv) = \mathsf{B}(u) + \mathsf{B}(v) = \mathsf{B}(vu)$.

For any word w from Σ^*, we denote by $\mathsf{Oc}(w)$ (resp. $\mathsf{Or}(w)$) the number of open calls (resp. open returns) in w. Formally,

$$\mathsf{Or}(w) = -\min\{\mathsf{B}(w') \mid w'w'' = w\} \qquad \mathsf{Oc}(w) = \mathsf{B}(w) + \mathsf{Or}(w)$$

We define, for any word w, $\mathsf{O}(w)$ as the pair $(\mathsf{Or}(w), \mathsf{Oc}(w)) \in \mathbb{N}^2$. Given $(n_1, n_2) \in \mathbb{N}^2$, we define $||(n_1, n_2)|| = \max(n_1, n_2)$. Note that, for a word w, we obtain $||\mathsf{O}(w)|| = \max\{\mathsf{Or}(w), \mathsf{Oc}(w)\}$ and $w \in \Sigma^*_{\mathsf{wn}}$ iff $||\mathsf{O}(w)|| = 0$, that is $\mathsf{O}(w) = (0, 0)$.

Given a word $w \in \Sigma^*$, we let $\mathsf{height}(w) = \max\{||\mathsf{O}(w_1)|| \mid w = w_1 w_2\}$ be the height of w. We denote by $|w|$ the length of w, defined as usual.

Definition 1. *For any two pairs (n_1, n_2) and (n'_1, n'_2) of naturals from $\mathbb{N}^2$, we define $(n_1, n_2) \oplus (n'_1, n'_2)$ as the pair*

$$\begin{cases} (n_1, n_2 - n'_1 + n'_2) & \textit{if } n_2 \geq n'_1 \\ (n_1 + n'_1 - n_2, n'_2) & \textit{if } n'_1 > n_2 \end{cases}$$

Proposition 1. $(\mathbb{N}^2, \oplus, (0,0))$ *is a monoid, and the mapping* O *is a morphism from* $(\Sigma^*, ., \epsilon)$ *to* $(\mathbb{N}^2, \oplus, (0,0))$*; in particular, for any two words u_1, u_2 from Σ^*, $O(u_1 u_2) = O(u_1) \oplus O(u_2)$.*

Transductions - Transducers Let Σ be a structured (input) alphabet, and Δ be a structured (output) alphabet. A relation over $\Sigma^* \times \Delta^*$ is a *transduction*. We denote by $\mathcal{T}(\Sigma, \Delta)$ the set of these transductions. For a transduction T, the set of words u (resp. v) such that $(u, v) \in T$ is called the *domain* (resp. the *range*) of T.

A *visibly pushdown transducer* from Σ to Δ (the class is denoted $\mathsf{VPT}(\Sigma, \Delta)$) is a tuple $A = (Q, I, F, \Gamma, \delta)$ where Q is a finite set of states, $I \subseteq Q$ the set of initial states, $F \subseteq Q$ the set of final states, Γ a (finite) stack alphabet, and $\delta = \delta_c \uplus \delta_r \uplus \delta_i$ is the transition relation where:

- $\delta_c \subseteq Q \times \Sigma_c \times \Gamma \times \Delta^* \times Q$ are the *call transitions*,
- $\delta_r \subseteq Q \times \Sigma_r \times \Gamma \times \Delta^* \times Q$ are the *return transitions*.
- $\delta_i \subseteq Q \times \Sigma_i \times \Delta^* \times Q$ are the *internal transitions*.

A stack (content) is a word over Γ. Hence, Γ^* is a monoid for the concatenation with $\perp$ (the empty stack) as neutral element. A configuration of A is a pair (q, σ) where $q \in Q$ and $\sigma \in \Gamma^*$ is a stack content. Let $u = a_1 \dots a_l$ be a (nested) word on Σ, and $(q, \sigma), (q', \sigma')$ be two configurations of A. A *run* of the VPT A over u from (q, σ) to (q', σ') is a (possibly empty) sequence of transitions $\rho = t_1 t_2 \dots t_l \in \delta^*$ such that there exist $q_0, q_1, \dots q_l \in Q$ and $\sigma_0, \dots \sigma_l \in \Gamma^*$ with $(q_0, \sigma_0) = (q, \sigma)$, $(q_l, \sigma_l) = (q', \sigma')$, and for each $0 < k \leq l$, we have either (i) $t_k = (q_{k-1}, a_k, \gamma, w_k, q_k) \in \delta_c$ and $\sigma_k = \sigma_{k-1}\gamma$, or (ii) $t_k = (q_{k-1}, a_k, \gamma, w_k, q_k) \in \delta_r$, and $\sigma_{k-1} = \sigma_k \gamma$, or (iii) $t_k = (q_{k-1}, a_k, w_k, q_k) \in \delta_i$, and $\sigma_{k-1} = \sigma_k$. When the sequence of transitions is empty, $(q, \sigma) = (q', \sigma')$.

The length (resp. height) of a run ρ over some word $u \in \Sigma^*$, denoted $|\rho|$ (resp. $\mathsf{height}(\rho)$) is defined as the length of u (resp. as the height of u).

The *output* of ρ (denoted $\mathsf{output}(\rho)$) is the word $v \in \Delta^*$ defined as the concatenation $w = w_1 \dots w_l$ when the sequence of transitions is not empty and ϵ otherwise. We write $(q, \sigma) \xrightarrow{u|w} (q', \sigma')$ when there exists a run on u from (q, σ) to (q', σ') producing w as output. Initial (resp. final) configurations are pairs $(q, \perp)$ with $q \in I$ (resp. with $q \in F$). A configuration (q, σ) is *reachable* (resp. *co-reachable*) if there exists some initial configuration $(i, \perp)$ (resp. some final configuration $(f, \perp)$) and a run from $(i, \perp)$ to (q, σ) (resp. from (q, σ) to $(f, \perp)$). A run is *accepting* if it starts in an initial configuration and ends in a final configuration.

A transducer A defines relation/transduction from nested words to nested words, denoted by $[\![A]\!]$, and defined as the set of pairs $(u, v) \in \Sigma^* \times \Delta^*$ such that there exists an accepting run on u producing v as output. Note that since both initial and final configurations have empty stack, A accepts only well-nested words, i.e. $[\![A]\!] \subseteq \Sigma^*_{\mathsf{wn}} \times \Delta^*$.

We denote $\mathcal{VP}(\Sigma, \Delta)$ the class of transductions defined by VPTs over the structured alphabets Σ (as input alphabet) and Δ (as output alphabet).

Given a VPT $A = (Q, I, F, \Gamma, \delta)$, we let $\mathsf{O}^A_{\max}$ be the maximal number of open calls and of open returns in a word produced as output of a call or of a return transition in A. Formally, we have:

$$\mathsf{O}^A_{\max} = \max\{||\mathsf{O}(w)|| \mid (p, \alpha, w, \gamma, q) \in \delta_c \cup \delta_r\}$$

Visibly pushdown automata We define visibly pushdown automata (VPA) simply as a particular case of VPT; we may think of them as transducers with no output. Hence, only the domain of the transduction matters and is called the language defined by the visibly pushdown automaton. For an automaton A, this language will be denoted $L(A)$.

Properties of computations in VPA/VPT We recall two standard results on runs of visibly pushdown machines.

Lemma 2. *Let A be a* VPA *with set of states Q and $\rho : (p, \perp) \xrightarrow{u} (q, \perp)$ be a run of A over some word $u \in \Sigma^*_{\mathsf{wn}}$. Let $h \in \mathbb{N}_{>0}$. We have:*

(i) *if* $\mathsf{height}(u) < h$ *and* $|u| \geq |Q|^h$, *then ρ can be decomposed as follows:*

$$\rho : (p, \perp) \xrightarrow{u_1} (p_1, \sigma) \xrightarrow{u_2} (p_1, \sigma) \xrightarrow{u_3} (q, \perp)$$

with u_1u_3 and u_2 well-nested words and $u_2 \neq \epsilon$.

(ii) *if* $\mathsf{height}(u) \geq |Q|^2$, *then ρ can be decomposed as follows:*

$$\rho : (p, \perp) \xrightarrow{u_1} (p_1, \sigma) \xrightarrow{u_2} (p_1, \sigma\sigma') \xrightarrow{u_3} (p_2, \sigma\sigma') \xrightarrow{u_4} (p_2, \sigma) \xrightarrow{u_5} (q, \perp)$$

with u_1u_5, u_2u_4 and u_3 well-nested words, and $\sigma' \neq \perp$.

3 Classes of VPT producing (almost) well-nested outputs

In this section, after recalling the definition of (locally) well-nested VPT, we introduce the new classes of globally and almost well-nested VPT. Then, we prove relationships between these classes.

3.1 Definitions

Locally Well-nested VPT*s (*lwnVPT*)* In [6], the class of (locally) well-nested VPT has been introduced. For this class, the enforcement of the well-nestedness of the output is done locally and syntactically at the level of transition rules.

Definition 2 (Locally Well-nested). *Let $A = (Q, I, F, \Gamma, \delta)$ be a* VPT*. A is a* locally well-nested VPT (lwnVPT) *if:*

- *for any pair of transitions $(q, a, v, \gamma, q') \in \delta_c$, $(p, b, w, \gamma, p') \in \delta_r$, the word vw is well nested, and*
- *for any transition $(q, a, v, q') \in \delta_i$, the word v is well-nested.*

A VPT *transduction T is* locally well-nested *if there exists a* lwnVPT *A that realizes T ($[\![A]\!] = T$). The class of locally well-nested* VPT *transductions is denoted $\mathcal{L}_{\mathsf{wn}}$.*

It is straightforward to prove that

Proposition 2. *Let A be a locally well-nested* VPT *and (p, σ), (q, σ) two configurations of A. For all well-nested word u, if $(p, \sigma) \xrightarrow{u/v} (q, \sigma)$ then $v \in \Sigma^*_{\mathsf{wn}}$.*

Therefore, any locally well-nested VPT transduction T is included into $\Sigma^*_{\mathsf{wn}} \times \Delta^*_{\mathsf{wn}}$.

Globally well-nested VPT *transduction - Almost well-nested* VPT *transduction* In this section, we introduce the class of globally well-nested transductions and its weaker variant of "almost" well-nested transductions. Unlike the definition of $\mathcal{L}_{\mathsf{wn}}$ which is done at the level of transducers, these definitions are done at the level of transductions and thus, as a semantical property.

Definition 3 (Globally Well-nested). *A* VPT *transduction T is* globally well-nested *if $T(\Sigma^*_{\mathsf{wn}}) \subseteq \Delta^*_{\mathsf{wn}}$. The class of globally well-nested* VPT *transductions is denoted $\mathcal{G}_{\mathsf{wn}}$.*

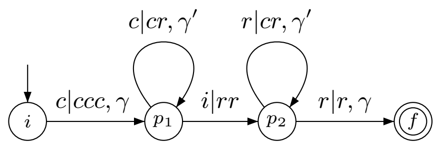
A VPT *A is* globally well-nested *if its transduction $[\![A]\!]$ is. The class of globally well-nested* VPT *is denoted* gwnVPT.

Definition 4 (Almost Well-nested). *A* VPT *transduction T is* almost well-nested *if there exists k in $\mathbb{N}$ such that for every pair of words $(u, v) \in T$, it holds that $||\mathsf{O}(v)|| \leq k$. The class of almost well-nested* VPT *transductions is denoted $\mathcal{A}_{\mathsf{wn}}$.*

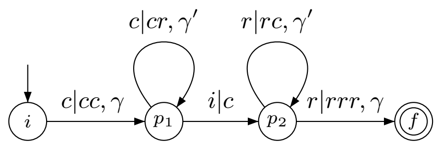
A VPT *A is* almost well-nested *if its transduction $[\![A]\!]$ is. The class of almost well-nested* VPT *is denoted* awnVPT.

3.2 Comparison of the different classes

Classes of transductions $\mathcal{G}_{\mathsf{wn}}$ and $\mathcal{A}_{\mathsf{wn}}$ are defined by semantical conditions on the defined relations. This yields a clear correspondence between the classes $\mathcal{G}_{\mathsf{wn}}$ and gwnVPT on one side and $\mathcal{A}_{\mathsf{wn}}$ and awnVPT on the other side. This is not the case for $\mathcal{L}_{\mathsf{wn}}$: two examples of VPTs are given in Figure 1. It is easy to verify that $A_1, A_2 \in \mathsf{gwnVPT}$. Moreover, none of these transducers belongs to lwnVPT. However, one can easily build a transducer A'_2 such that $[\![A_2]\!] = [\![A'_2]\!]$ and $A'_2 \in \mathsf{lwnVPT}$. Indeed one can perform the following modifications:



(a) The VPT A_1.



(b) The VPT A_2.

Fig. 1. Two VPTs in $\mathcal{VP}(\Sigma, \Sigma)$ with $\Sigma_c = \{c\}$, $\Sigma_r = \{r\}$ and $\Sigma_i = \{i\}$.

- the transition (p_1, i, c, p_2) becomes (p_1, i, ϵ, p_2)
- the transition $(p_2, r, rc, \gamma', p_2)$ becomes $(p_2, r, cr, \gamma', p_2)$
- the transition (p_2, r, rrr, γ, f) becomes $(p_2, r, crrr, \gamma, f)$

On the contrary, as we prove below, the transduction $[\![A_1]\!]$ does not belong to $\mathcal{L}_{\mathsf{wn}}$: there exists no transducer $A_1' \in \mathsf{lwnVPT}$ such that $[\![A_1']\!] = [\![A_1]\!]$.

To summarize, we prove the following proposition.

Proposition 3. *The following inclusion results hold:*

- *For transducers:* $\mathsf{lwnVPT} \subsetneq \mathsf{gwnVPT} \subsetneq \mathsf{awnVPT}$
- *For transductions:* $\mathcal{L}_{\mathsf{wn}} \subsetneq \mathcal{G}_{\mathsf{wn}} \subsetneq \mathcal{A}_{\mathsf{wn}}$

Proof (Sketch). The non-strict inclusions are straightforward. The two strict inclusions $\mathsf{gwnVPT} \subsetneq \mathsf{awnVPT}$ and $\mathcal{G}_{\mathsf{wn}} \subsetneq \mathcal{A}_{\mathsf{wn}}$ follow from the constraint on the range. The strict inclusion $\mathsf{lwnVPT} \subsetneq \mathsf{gwnVPT}$ is witnessed by A_2 from Figure 1, as explained above.

We sketch now the proof of the strict inclusion $\mathcal{L}_{\mathsf{wn}} \subsetneq \mathcal{G}_{\mathsf{wn}}$, and therefore consider the transducer A_1 on Figure 1. Observe that $[\![A_1]\!] \in \mathcal{G}_{\mathsf{wn}}$, we show that $[\![A_1]\!] \notin \mathcal{L}_{\mathsf{wn}}$. First note that $[\![A_1]\!] = \{(cc^k ir^k r, ccc(cr)^k rr(cr)^k r) \mid k \in \mathbb{N}\}$ and that

- (Fact 1) The transduction defined by A_1 is injective
- (Fact 2) Any word of the output can be decomposed as $w_1 rrw_2$ where $w_1 = ccc(cr)^k$ and $w_2 = (cr)^k r$ for some natural k and for each w_1 with fixed k there exists a unique w_2 such that $w_1 rrw_2$ is in the range of A_1 (and conversely).

By contradiction, suppose that there exists $A_1' \in \mathsf{lwnVPT}$ such that $[\![A_1']\!] = [\![A_1]\!]$. Now, for k sufficiently large and depending only on the fixed size of A_1', A_1' has an accepting run for the input $cc^k ir^k r$ of the form given in the point (ii) of Lemma 2. Let us denote by u_i (resp. v_i), $i \in \{1, \ldots, 5\}$ the corresponding decomposition of the input (resp. output) word. Due to Proposition 2, words $v_1 v_5$, $v_2 v_4$ and v_3 are well-nested.

Now, assume that $v_2 = \epsilon$ and $v_4 = \epsilon$. Then, using a simple pumping argument over the pair (u_2, u_4), one would obtain a different input producing the same output, contradicting the injectivity of A_1' (due to (Fact 1)). So, $v_2 \neq \epsilon$ or $v_4 \neq \epsilon$.

Using a case analysis on the presence of the previously mentioned pattern rr in the outputs of A_1', using the fact that $v_2 v_4 \neq \epsilon$, (Fact 2) and a pumping argument over the pair of words (u_2, u_4), one obtains a contradiction. $\square$

4 Characterizations

In this section we give criteria on VPTs that aim to characterize the classes gwnVPT and awnVPT.

Definition 5. *Let A be a* VPT. *Let us consider the following criteria:*

(C1) For all states p, i, f such that i is initial and f is final, for any stack σ, then any accepting run

$$(i, \bot) \xrightarrow{u_1/v_1} (p, \sigma) \xrightarrow{u_2/v_2} (p, \sigma) \xrightarrow{u_3/v_3} (f, \bot)$$

with $u_1u_3, u_2 \in \Sigma^*_{\mathsf{wn}}$ satisfies $\mathsf{B}(v_2) = 0$.

(C2) For all states p, q, i, f such that i is initial and f is final, for any stack σ, σ', then any accepting run

$$(i, \bot) \xrightarrow{u_1/v_1} (p, \sigma) \xrightarrow{u_2/v_2} (p, \sigma\sigma') \xrightarrow{u_3/v_3} (q, \sigma\sigma') \xrightarrow{u_4/v_4} (q, \sigma) \xrightarrow{u_5/v_5} (f, \bot)$$

with $u_2u_4, u_3 \in \Sigma^*_{\mathsf{wn}}$ and $\sigma' \neq \bot$ satisfies $\mathsf{B}(v_2) + \mathsf{B}(v_4) = 0$ and $\mathsf{B}(v_2) \geq 0$.

The following result follows from Propositions 4 and 5 that we prove below.

Theorem 1. *A* VPT *A is almost well-nested iff it verifies criteria (C_1) and (C_2).*

Lemma 3. *Let $X \subseteq \Sigma^*$ such that the set $\mathsf{B}(X) = \{\mathsf{B}(u) \mid u \in X\}$ is infinite. Then the set $\{\mathsf{O}(u) \mid u \in X\}$ is infinite as well.*

Lemma 4. *Let $u \in \Sigma^*$ and k be a strictly positive integer. Then $\mathsf{O}(u^k)$ is equal to* $(\mathsf{Or}(u), (\mathsf{Oc}(u) - \mathsf{Or}(u)) * (k-1) + \mathsf{Oc}(u))$ *if* $\mathsf{Oc}(u) \geq \mathsf{Or}(u)$ *and to* $(\mathsf{Or}(u) + (\mathsf{Or}(u) - \mathsf{Oc}(u)) * (k-1), \mathsf{Oc}(u))$ *otherwise.*

Proof. By definition of $\oplus$ and by induction on k. □

Proposition 4. *Let A be a* VPT. *If A does not satisfy $(C1)$ or $(C2)$, then $A \notin$* awnVPT.

Proof. Let us assume that A does not satisfy $(C1)$. Hence there exists an accepting run as described in criterion $(C1)$ such that $\mathsf{B}(v_2) \neq 0$. We then build by iterating the loop on word u_2 accepting runs for words of the form $u_1(u_2)^k u_3$ for any natural k, producing output words $v_1(v_2)^k v_3$. Let us denote this set by X. As $\mathsf{B}(v_2) \neq 0$ and by Lemma 1, the set $\mathsf{B}(X)$ is infinite. Lemma 3 entails that A is not almost well-nested.

Assume now that A does not satisfy $(C2)$. Hence, there exists an accepting run as described in the statement of the proposition such that either (i) $\mathsf{B}(v_2) + \mathsf{B}(v_4) = b \neq 0$ or (ii) $\mathsf{B}(v_2) < 0$. In the case of (i), from this run, one can build by pumping accepting runs for words of the form $u_1(u_2)^k u_3 (u_4)^k u_5$ for any natural k, producing output words $v_1(v_2)^k v_3 (u_4)^k v_5$. As before, Lemmas 1 and 3 imply that A is not almost well-nested.

Now, for (ii) assuming that $\mathsf{B}(v_2) + \mathsf{B}(v_4) = 0$. As $\mathsf{B}(v_2) < 0$, it holds that $\mathsf{B}(v_4) > 0$ and thus, $\mathsf{Or}(v_2) > \mathsf{Oc}(v_2)$, $\mathsf{Or}(v_4) < \mathsf{Oc}(v_4)$. From the run of the statement, one can build by pumping accepting runs for words of the form $u_1(u_2)^k u_3 (u_4)^k u_5$ for any natural k, producing output words $v_1(v_2)^k v_3 (v_4)^k v_5$. Now, we consider $\mathsf{O}(v_1(v_2)^k v_3 (v_4)^k v_5)$

which, by associativity of $\oplus$, is equal to $\mathsf{O}(v_1)\oplus\mathsf{O}((v_2)^k)\oplus\mathsf{O}(v_3)\oplus\mathsf{O}((u_4)^k)\oplus\mathsf{O}(v_5))$. Now, by Lemma 4, it is equal to

$$\mathsf{O}(v_1) \oplus (\mathsf{Or}(v_2) + (\mathsf{Or}(v_2) - \mathsf{Oc}(v_2)) * (k-1), \mathsf{Oc}(v_2)) \oplus \mathsf{O}(v_3) \oplus \\ (\mathsf{Or}(v_4), (\mathsf{Oc}(v_4) - \mathsf{Or}(v_4)) * (k-1) + \mathsf{Oc}(v_4)) \oplus \mathsf{O}(v_5)$$

It is easy to see that for k varying, the described pairs are unbounded. □

Given a VPT $A = (Q, I, F, \Gamma, \delta)$, we define the integer $N_A = 2|Q|^{2|Q|^2}$.

Lemma 5. *Let A be a* VPT. *If A satisfies the criteria $(C1)$ and $(C2)$, then for any accepting run ρ such that $|\rho| \geq N_A$, there exists an accepting run ρ' such that $|\rho'| < |\rho|$ and* $||\mathsf{O}(\mathsf{output}(\rho'))|| \geq ||\mathsf{O}(\mathsf{output}(\rho))||$.

Proof (Sketch). Let $A = (Q, I, F, \Gamma, \delta)$ and ρ be an accepting run such that $|\rho| \geq N_A$. We distinguish two cases, depending on $\mathsf{height}(\rho)$:

- when $\mathsf{height}(\rho) < 2|Q|^2$: by definition of N_A, we can apply Lemma 2.(i) twice and prove that ρ is of the following form:

  $$(i, \bot) \xrightarrow{u_1/v_1} (p, \sigma) \xrightarrow{u_2/v_2} (p, \sigma) \xrightarrow{u_3/v_3} (q, \sigma') \xrightarrow{u_4/v_4} (q, \sigma') \xrightarrow{u_5/v_5} (f, \bot)$$

  with $u_2, u_4 \in \Sigma^*_{\mathsf{wn}} \setminus \{\epsilon\}$. Then, by criterion $(C1)$, we have $\mathsf{B}(v_2) = \mathsf{B}(v_4) = 0$. One can prove that at least one of u_2 and u_4 can be removed from u while preserving the value $\mathsf{Or}(u)$. Let us denote by v' the resulting output word. Observe also that removing this part of the run does not modify the balance $\mathsf{B}(.)$ of the run, as $\mathsf{B}(v_2) = \mathsf{B}(v_4) = 0$. As $\mathsf{Oc}(v) = \mathsf{B}(v) + \mathsf{Or}(v)$, we obtain $\mathsf{O}(v) = \mathsf{O}(v')$, yielding the result.
- when $\mathsf{height}(\rho) \geq 2|Q|^2$: in this case, we can apply Lemma 2.(ii) twice and prove that ρ is of the following form:
  $(i, \bot) \xrightarrow{u_1/v_1} (p_1, \sigma) \xrightarrow{u_2/v_2} (p_1, \sigma\sigma_1) \xrightarrow{u_3/v_3} (q_1, \sigma\sigma_1\sigma_2) \xrightarrow{u_4/v_4} (q_1, \sigma\sigma_1\sigma_2\sigma_3)$ $\xrightarrow{u_5/v_5} (q_2, \sigma\sigma_1\sigma_2\sigma_3) \xrightarrow{u_6/v_6} (q_2, \sigma\sigma_1\sigma_2) \xrightarrow{u_7/v_8} (p_2, \sigma\sigma_1) \xrightarrow{u_8/v_8} (p_2, \sigma) \xrightarrow{u_9/v_9}$ $(f, \bot)$, with $u_1u_9, u_2u_8, u_3u_7, u_4u_6, u_5 \in \Sigma^*_{\mathsf{wn}}$ and $\sigma_1, \sigma_3 \neq \bot$.
  Then the two following runs can be built: the one obtained by removing the parts of ρ on u_2 and u_8, and the one obtained by removing the parts of ρ on u_4 and u_6, yielding runs whose length is strictly smaller than $|\rho|$. Let us denote these two runs by ρ' and ρ'' respectively, and their outputs by v' and v''. As A verifies the criterion $(C2)$, we have that $\mathsf{B}(v) = \mathsf{B}(v') = \mathsf{B}(v'')$, as $\mathsf{B}(v_2)+\mathsf{B}(v_8) = \mathsf{B}(v_4)+\mathsf{B}(v_6) = 0$ and B is commutative. In order to obtain the result, we study $\mathsf{Or}(v)$. Considering different cases, we manage to prove that either $\mathsf{Or}(v') \geq \mathsf{Or}(v)$ or $\mathsf{Or}(v'') \geq \mathsf{Or}(v)$. The result follows as for any word w we have $\mathsf{Oc}(w) = \mathsf{B}(w) + \mathsf{Or}(w)$. □

Proposition 5. *Let A be a* VPT. *If A satisfies $(C1)$ and $(C2)$, then every accepting run $\rho : (i, \bot) \xrightarrow{u|v} (f, \bot)$ of A verifies* $||\mathsf{O}(v)|| \leq N_A.\mathsf{O}^A_{\max}$.

Proof. If $|\rho| \leq N_A$ the result is trivial; otherwise, assuming the existence of a minimal counter-example of this statement, a contradiction follows from Lemma 5. □

Now we can show a precise characterization of transducers from gwnVPT amongst those in awnVPT.

Definition 6. *Let A be a* VPT. *We consider the following criterion:*

(D) For all $(u, v) \in [\![T]\!]$, if $|u| \leq N_A$ then $v \in \Sigma^*_{\mathsf{wn}}$.

Theorem 2. *A* VPT *A is globally well-nested iff it verifies criteria (C_1), (C_2) and (D).*

Proof. The direct implication is trivial, the other one follows from Lemma 5. □

5 Deciding the classes of almost and globally well-nested VPT

In this section, we prove that given a VPT A, it is decidable to know whether $[\![A]\!] \in \mathcal{A}_{\mathsf{wn}}$ and whether $[\![A]\!] \in \mathcal{G}_{\mathsf{wn}}$. It is known that

Proposition 6. *Given a* VPT *$A = (Q, I, F, \Gamma, \delta)$ and states p, q of A, deciding whether there exists some stack σ such that (p, σ) is reachable and (q, σ) is co-reachable can be done in* PTIME.

Theorem 3. *Let A be a* VPT. *Whether $[\![A]\!] \in \mathcal{A}_{\mathsf{wn}}$ can be decided in* PSPACE.

Proof (Sketch). By Theorem 1, deciding the class awnVPT amounts to decide criteria $(C1)$ and $(C2)$. Therefore we propose a non-deterministic algorithm running in polynomial space, yielding the result thanks to Savitch theorem.

We claim that A verifies $(C1)$ and $(C2)$ if and only if it verifies these criteria on "small instances", defined as follows:

- Criterion $(C1)$: consider only words u_2 such that $\mathsf{height}(u_2) \leq |Q|^2$ and $|u_2| \leq 2.|Q|^{|Q|^2}$.
- Criterion $(C2)$: consider only stacks σ' such that $|\sigma'| \leq |Q|^2$ and words u_2, u_4 of height at most $2.|Q|^2$ and length at most $|Q|^2.|Q|^{|Q|^2}$.

The non-deterministic algorithm follows from the claim: in order to exhibit a witness of the fact that $A \notin \mathsf{awnVPT}$, the algorithm guesses whether $(C1)$ or $(C2)$ is violated; then, the claim implies the existence of a witness of at most exponential size. This witness can be guessed on-the-fly in polynomial space. Proposition 6 is then used to check that the witness can be completed into an accepted run.

To prove this claim, we show, by induction on $u \in \Sigma^*_{\mathsf{wn}}$, that for every run $(p, \perp) \xrightarrow{u|v} (q, \perp)$ that can be completed into an accepting run, and for every decomposition of this run according to criterion (C_1) or $(C2)$, the property stated by the corresponding criterion is fulfilled. □

The previous algorithm could be extended to handle in addition criterion (D), yielding a PSPACE algorithm to decide whether a VPT A is globally well-nested. However, we can use a recent result to prove that this problem can be solved in PTIME.

Theorem 4. *Let A be a* VPT. *Whether $[\![A]\!] \in \mathcal{G}_{\mathsf{wn}}$ can be decided in* PTIME.

Proof. This proof relies on results from [2] showing that deciding whether a context-free language is included into a Dyck language can be solved in PTIME.

We first erase the precise symbols of the produced outputs keeping track only of the type of the symbols: we build from A a VPT A' defined on the structured output alphabet Σ' with $\Sigma'_c = \{(\}$, $\Sigma'_r = \{)\}$ and $\Sigma'_i = \emptyset$. A transition of A' is obtained from a transition of A by replacing in output words of the transition of A call symbols by (and return symbols by) and removing internal symbols. It is then easy to see that A is in gwnVPT iff A' is in gwnVPT (actually, for each run in A producing v, its corresponding run in A' produces some v' such that $\mathsf{O}(v) = \mathsf{O}(v')$). Then, as shown in [9], one can build in polynomial time a context-free grammar $G_{A'}$ generating the range of A'. Finally, we appeal to [2] to conclude. □

6 Closure under composition and Type-checking

6.1 Definitions and existing results

We consider two natural problems for transducers : the first one is related to composition of transductions. The second problem is the type-checking problem that aims to verify that any output of a transformation belongs to some given type/language. For VPT, the obvious class of "types" to consider is the class of languages defined by VPA.

Definition 7 (Closure under composition). *A class $\mathcal{T}$ of transductions included in $\Sigma^* \times \Sigma^*$ is closed under composition if for all T, T' in $\mathcal{T}$, the transduction $T \circ T'$ is also in $\mathcal{T}$. It is effectively closed under composition if for any transducers A, A' such that $[\![A]\!], [\![A']\!] \in \mathcal{T}$, $A \circ A'$ is computable and $[\![A \circ A']\!]$ is in $\mathcal{T}$.*

A class of transducers T *is effectively closed under composition if for any two transducers A, A' in* T*, $A \circ A'$ is computable and $A \circ A'$ is in* T.

Definition 8 (Type-checking (against VPA)). *Given a* VPT *A and two VPA B, C, decide whether $[\![A]\!](L(B)) \subseteq L(C)$.*

The following results give the status of these properties for arbitrary VPTs and for lwnVPT:

Theorem 5 ([7,6]). *Regarding closure under composition, we have:*

- *The class $\mathcal{VP}(\Sigma, \Sigma)$ is not closed under composition.*
- *The class* lwnVPT *is effectively closed under composition.*

In addition, the problem of type checking against VPA *is undecidable for (arbitrary)* VPT *and decidable for* lwnVPT.

6.2 New results

Actually, regarding the closure under composition of the class lwnVPT, though not explicitly stated, the result proved in [6] is slightly stronger. It is indeed shown that for any VPT A, B such that $A \in$ lwnVPT, there exists an (effectively computable) VPT C satisfying $[\![C]\!] = [\![A]\!] \circ [\![B]\!]$. In addition, if $B \in$ lwnVPT, then $C \in$ lwnVPT.

We extend this positive result to any almost well-nested transducer.

One of the main ingredients of the proof of this result is the set $\mathcal{UPS}_A$ defined for any VPT transducer $A = (Q_A, I_A, F_A, \Gamma_A, \delta^A)$ as

$$\left\{ (p, p', n_1, n_2) \middle| \begin{array}{l} \exists \sigma \in \Gamma^*,\ (p, \sigma) \text{ is reachable and } (p', \sigma) \text{ is co-reachable and} \\ \exists u \in \Sigma^*_{\mathsf{wn}}, (p, \bot) \xrightarrow{u|v} (p', \bot) \text{ and } \mathsf{O}(v) = (n_1, n_2) \end{array} \right\}$$

Proposition 7. *Let A in* awnVPT. *Then the set $\mathcal{UPS}_A$ is finite and computable in exponential time in the size of A.*

Theorem 6. *Let A, B be two* VPT*s. If A is almost-well nested, then one can compute in exponential time in the size of A and B a* VPT *C such that $[\![C]\!] = [\![A]\!] \circ [\![B]\!]$. Moreover, if B is also almost well-nested, then so is C, and if A and B are globally well-nested, then so is C.*

Proof (Sketch). We present the construction of C. By Proposition 7, $\mathcal{UPS}_A$ is finite and we let K be the computable integer value $\max\{||(n_1, n_2)|| \mid (p, p', n_1, n_2) \in \mathcal{UPS}_A\}$.

Given $B = (Q_B, I_B, F_B, \Gamma_B, \delta^B)$, we define $C = (Q_C, I_C, F_C, \Gamma_C, \delta^C)$ as

$$\begin{array}{ll} Q_C = Q_A \times Q_B \times \Gamma_B^{\leq K} & I_C = I_A \times I_B \times \{\bot\} \\ \Gamma_C = \Gamma_A \times \Gamma_B^{\leq \mathsf{O}^A_{\max} + K} & F_C = F_A \times F_B \times \{\bot\} \end{array}$$

Now for the transition rules δ^C:

- $((p, q, \sigma), i, w, (p', q', \sigma')) \in \delta^C_i$ if there exist a word $v \in \Delta^*$ and a stack $\sigma_0 \in \Gamma^*_B$ such that $\sigma = \sigma_0\sigma_1$, $\sigma' = \sigma_0\sigma'_1$, $\mathsf{O}(v) = (|\sigma_1|, |\sigma'_1|)$, and $(p, i, v, p') \in \delta^A_i$ and there exists a run $(q, \sigma_1) \xrightarrow{v|w} (q', \sigma'_1)$ in B,
- $((p, q, \sigma), c, w, (\gamma, \sigma_3), (p', q', \sigma_4)) \in \delta^C_c$ if there exist a word $v \in \Delta^*$, two stacks $\sigma_0, \sigma_2 \in \Gamma^*_B$ and a stack symbol $\gamma \in \Gamma_A$ such that $\sigma = \sigma_0\sigma_1$, $\mathsf{O}(v) = (|\sigma_1|, |\sigma_2|)$, $\sigma_0\sigma_2 = \sigma_3\sigma_4$, $(p, c, v, \gamma, p') \in \delta^A_c$ and there exists a run $(q, \sigma_1) \xrightarrow{v|w} (q', \sigma_2)$ in B such a transition exists provided the bounds on the sizes of the different stacks are fulfilled, *i.e.* $|\sigma| \leq K$, $|\sigma_4| \leq K$, and $|\sigma_3| \leq \mathsf{O}^A_{\max} + K$,
- $((p, q, \sigma), r, w, (\gamma, \sigma_3), (p', q', \sigma')) \in \delta^C_r$ if there exist a word $v \in \Delta^*$, a stack $\sigma_0 \in \Gamma^*_B$ such that $\sigma_0\sigma_1 = \sigma_3\sigma$, $\sigma_0\sigma_2 = \sigma'$, $\mathsf{O}(v) = (|\sigma_1|, |\sigma_2|)$, $(p, r, v, \gamma, p') \in \delta^A_r$ and there exists a run $(q, \sigma_1) \xrightarrow{v|w} (q', \sigma_2)$ in B such a transition exists provided the bounds on the sizes of the different stacks are fulfilled, *i.e.* $|\sigma| \leq K$, $|\sigma'| \leq K$, and $|\sigma_3| \leq \mathsf{O}^A_{\max} + K$.

In a state of C, we store the current states of A and B. In addition, a part of the top of the stack of B is also stored in the state of C to allow the simulation of B. The (finite) amount that needs to be stored in the state is identified using the set $\mathcal{UPS}_A$. □

Corollary 1. *The classes $\mathcal{G}_{\mathsf{wn}}$ and $\mathcal{A}_{\mathsf{wn}}$ are (effectively) closed under composition.*

Theorem 7 (Type-checking against VPA**).** *Given an almost well-nested* VPT *A and two visibly pushdown automata B, C, whether $[\![A]\!](L(B)) \subseteq L(C)$ is decidable in* 2 − EXPTIME.

Proof. Restricting the domain of A to $L(B)$ is easy: it suffices to compute the product VPA of A and B. Then, VPA being closed under complementation, we compute $\overline{C}$, the complement of C. Note that the size of $\overline{C}$ is at most exponential in the size of C. We then turn $\overline{C}$ into a transducer C' defining the identity relation over $L(\overline{C})$ (this is obvious by simply transforming rules of $\overline{C}$ into rules of transducers outputting their input). Now, by Theorem 6, one can build a transducer defining the composition of $[\![A]\!] \circ [\![C']\!]$. This can be done in doubly exponential time in the size of A and C. Now, it is sufficient to test whether the VPA underlying this transducer is empty or not. $\square$

7 Conclusion

In this paper, we have considered and precisely characterized the class of VPT with well-nested outputs. We have shown that this class is closed under composition and that its type-checking against VPA is decidable. We have restricted ourselves in this paper to transducers with well-nested domains. We conjecture that this restriction can be easily relaxed and thus, one could consider transducers based on nested word automata [1]. We left open the problem of deciding the class $\mathcal{L}_{\mathsf{wn}}$. As we have described on some examples, this problem is far from being trivial. In [4], a clear relationship between the class lwnVPT and hedge-to-hedge transducers is described; investigating such a relationship for gwnVPT is also an interesting problem.

References

1. R. Alur and P. Madhusudan. Adding Nesting Structure to Words. *Journal of the ACM*, 56(3):1–43, 2009.
2. A. Bertoni, C. Choffrut, and R. Radicioni. The inclusion problem of context-free languages: Some tractable cases. *International Journal of Foundations of Computer Science*, 22(2):289–299, 2011.
3. B. v. Braunmühl and R. Verbeek. Input-driven Languages are Recognized in log n Space. In *Fundamentals of Computation Theory, 4th International Conference*, volume 158 of *LNCS*, pages 40–51, 1983.
4. M. Caralp, E. Filiot, P.-A. Reynier, F. Servais, and J.-M. Talbot. Expressiveness of visibly pushdown transducers. In *Second International Workshop on Trends in Tree Automata and Tree Transducers*, volume 134 of *EPTCS*, pages 17–26, 2013.
5. E. Filiot, O. Gauwin, P.-A. Reynier, and F. Servais. Streamability of Nested Word Transductions. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 13 of *LIPIcs*, pages 312–324, 2011.
6. E. Filiot, J.-F. Raskin, P.-A. Reynier, F. Servais, and J.-M. Talbot. Properties of Visibly Pushdown Transducers. In *Mathematical Foundations of Computer Science 2010, 35th International Symposium*, volume 6281 of *LNCS*, pages 355–367, 2010.
7. J.-F. Raskin and F. Servais. Visibly pushdown transducers. In *Automata, Languages and Programming, 35th International Colloquium*, volume 5126 of *LNCS*, pages 386–397, 2008.
8. P.-A. Reynier and J.-M. Talbot. Visibly Pushdown Transducers with Well-nested Outputs. Technical report, http://hal.archives-ouvertes.fr/hal-00988129/PDF/wnVPT.pdf, 2014.
9. F. Servais. *Visibly Pushdown Transducers*. PhD thesis, Université Libre de Bruxelles, 2011.
10. S. Staworko, G. Laurence, A. Lemay, and J. Niehren. Equivalence of deterministic nested word to word transducers. In *Fundamentals of Computation Theory, 17th International Symposium*, volume 5699 of *LNCS*, pages 310–322, 2009.