Antichains Algorithms for the Inclusion Problem Between ω-VPL*

Kyveli Doveri[1,2](✉), Pierre Ganty[1], and Luka Hadži-Đokić[1]

[1] IMDEA Software Institute, Madrid, Spain
{kyveli.doveri,pierre.ganty,luka.hadzi-dokic}@imdea.org
[2] Universidad Politécnica de Madrid, Madrid, Spain

Abstract. We define novel algorithms for the inclusion problem between two visibly pushdown languages of infinite words, an EXPTIME-complete problem. Our algorithms search for counterexamples to inclusion in the form of ultimately periodic words i.e. words of the form uv^ω where u and v are finite words. They are parameterized by a pair of quasiorders telling which ultimately periodic words need not be tested as counterexamples to inclusion without compromising completeness. The pair of quasiorders enables distinct reasoning for prefixes and periods of ultimately periodic words thereby allowing to discard even more words compared to using the same quasiorder for both. We put forward two families of quasiorders: the state-based quasiorders based on automata and the syntactic quasiorders based on languages. We also implemented our algorithm and conducted an empirical evaluation on benchmarks from software verification.

1 Introduction

Visibly pushdown languages [4] (VPL) have applications in various domains including verification [22], theorem proving [27] or XML schema languages reasoning [26] where the inclusion problem plays a crucial role. For instance proving correctness relative to a specification reduces to a language inclusion problem and so does proving correctness of a theorem of the form $\forall x \exists y P(x) \implies Q(y)$. The extension to the case of visibly pushdown languages of infinite words (ω-VPL) has also been studied in the context of program verification [21] and it has applications in word combinatorics [23,25,27].

We distinguish two general approaches to solve the language inclusion problem $L \subseteq M$: (i) complement M, intersect with L and check for emptiness of the result; and (ii) reduce the inclusion check to finitely many *membership queries* asking whether $w \in M$ holds where $w \in L$ and each query aims at finding a counterexample to inclusion.

* This work was partially funded by the ESF Investing in your future, the RYC-2016-20281/MCIN/AEI/10.13039/501100011033, the Madrid regional government as part of the program S2018/TCS-4339 (BLOQUES-CM) co-funded by EIE Funds of the European Union, the PRODIGY Project (TED2021-132464B-I00) funded by MCIN and the European Union NextGenerationEU/ PRTR.

S. Sankaranarayanan and N. Sharygina (Eds.): TACAS 2023, LNCS 13993, pp. 290–307, 2023.
https://doi.org/10.1007/978-3-031-30823-9_15

In this paper we focus on the second approach. Previous work in that space leverage relations between words to select a finite subset of words of L on which we run the membership queries. A class of relations that consistently yields good results in practice are *quasiorders* which discard words subsumed (for the quasiorder) by others. A key feature of such quasiorders is that the subset of L selected via the quasiorder must contain a counterexample to inclusion if there exists one. Quasiorders are a versatile heuristic that has been applied to inclusion problems for languages such as languages of finite words [3,10,14] (including visibly pushdown language [6]) or infinite words [1,2,12,13,16,24] and even tree languages [3,5]. Algorithms leveraging quasiorders are commonly referred to as *antichains algorithms*. Subsequent improvements (e.g. [2] improving [1]) often attempt at defining coarser quasiorders because they enable the selection of an even smaller subset of L.

Let us now turn to the inclusion problem between ω-VPL, an EXPTime-complete problem. For that problem the selection of words of L is limited to *ultimately periodic words*, i.e. words of the form uv^ω, where u and v are called *prefix* and *period* respectively. For an ultimately periodic word uv^ω subsumption (for a quasiorder) simply means subsumption of (u, v) relative to a pair $\leqq_1 \times \leqq_2$ of quasiorders on finite words. The quasiorders found in the literature [17,18] are all equivalences and are all such that $\leqq_1 = \leqq_2$.

In this paper, we propose a new family of algorithms for the inclusion problem between ω-VPL that leverages a subset of the ultimately periodic words, deemed *legitimate decompositions* and is parameterized by a pair of quasiorders and a decision procedure for the membership queries in M. We identify properties that such pair of quasiorders must satisfy so that the resulting algorithm actually decides the inclusion problem between two ω-VPL: (1) be decidable; (2) be well-quasiorders; (3) verify some monotonicity conditions w.r.t. word operations that are characteristic to ω-VPL and (4) satisfy a preservation property intuitively saying that a legitimate decomposition inside M cannot subsume a legitimate decomposition outside of M. We put forward two families of quasiorders satisfying (1) thru (4): the *state-based quasiorders* whose definition rely on a visibly pushdown automaton underlying M and the *syntactic quasiorders* whose definition is based solely on M. The syntactic orders are the "ideal" quasiorders in the sense they are the coarsest, hence they select the "smallest" subset of L. None of our quasiorders is symmetric, hence they are coarser than equivalences and in each and every pair we define the quasiorder on prefixes differs from the one on periods (i.e. $\leqq_1 \neq \leqq_2$). We further prove that when instantiated with the state-based quasiorders and with a state-based decision procedure for membership queries the resulting algorithm, which we call the *state-based algorithm*, has a runtime that matches the corresponding problem complexity.

Finally we implement the state-based algorithm and evaluate it on various benchmarks collected from Friedmann et al. [18] and from SV-COMP[3], the Software Verification competition. The empirical evaluation is carried out against Ultimate [21] which follows a complement, intersect and check for emptiness

[3] https://sv-comp.sosy-lab.org

approach. The preliminary conclusion of the empirical results is in favor of our approach as it scales up better.

Related Work. Bruyere et al. [6] proposed an antichain algorithm for the inclusion of VPL but they only tackle the problem for languages of finite words. The same limitation applies to Ganty et al. [19,20] where, moreover, they do not tackle the inclusion problem of VPL into VPL (the closest they tackle is CFL into regular). The extension from the finite to the infinite case was tackled in Doveri et al. [13] but they do not cover the case ω-VPL into ω-VPL (the closest they tackle is ω-CFL into ω-regular). Friedmann et al. [17,18] do tackle the ω-VPL into ω-VPL problem. However they do not leverage the full power of quasiorders (they use equivalence instead); they do not use distinct pruning techniques for prefix and periods; and they do not put forward syntactic quasiorders. A summary comparing our work (omegaVPLinc) with the closest works in the area is given at Table 1.

Table 1. Comparison of the closest work in the area based on the characteristics of the problem tackled (first two columns) and the techniques used (last three columns). N/A means non applicable, ○ means no support and ● means full support. The labels ω, VPL, qo, $\leqq_1 \neq \leqq_2$ and syntactic qo ask respectively whether the work thereof tackles the problem of infinite words, tackles the problem of VPL, leverage quasiorders, defines distinct quasiorders for prefixes and periods, and defines syntactic quasiorders.

	ω	VPL	qo	$\leqq_1 \neq \leqq_2$	syntactic qo
Bruyere et al. [6]	○	●	●	N/A	○
Ganty et al. [20]	○	○	●	N/A	●
Doveri et al. [13]	●	○	●	●	○
Friedmann et al. [18]	●	●	○	○	○
omegaVPLinc	●	●	●	●	●

2 Background

Fix $\Sigma \triangleq \Sigma_i \cup \Sigma_c \cup \Sigma_r$ an alphabet (a finite non empty set of symbols) comprising three disjoint alphabets. The set of finite words and the set of infinite words over Σ are denoted by Σ^* and Σ^ω respectively. We denote by ϵ the empty word and define $\Sigma^+ \triangleq \Sigma^* \setminus \{\epsilon\}$. Given a word $u = u_0 u_1 \cdots \in \Sigma^* \cup \Sigma^\omega$ we say that a position j where $j \in \mathbb{N}$, $j < |u|$ and $|u| \in \mathbb{N} \cup \{\omega\}$ is the length of u, is an *internal* (resp. *call*, resp. *return*) position if $u_j \in \Sigma_i$ (resp. $u_j \in \Sigma_c$, resp. $u_j \in \Sigma_r$).

Visibly Pushdown Languages. A Visibly Pushdown Automaton (VPA) over Σ is a tuple $\mathcal{A} = (Q, q_I, \Gamma, \delta, F)$, where Q is a finite set of states including an initial state $q_I \in Q$, $F \subseteq Q$ is the set of final states, Γ is the stack alphabet including a bottom-of-stack symbol $\perp$ and $\delta = \delta_i \cup \delta_c \cup \delta_r$ consists of three transition relations $\delta_i \subseteq Q \times \Sigma_i \times Q$, $\delta_c \subseteq Q \times \Sigma_c \times Q \times \Gamma \setminus \{\perp\}$ and $\delta_r \subseteq Q \times \Sigma_r \times \Gamma \times Q$. *Configurations* in $\mathcal{A}$ are pairs in $Q \times \Gamma^*$. For $a \in \Sigma$ we define the relation $\vdash^a$ between configurations as follows:

– If $a \in \Sigma_i$ and $w \in \Gamma^*$ we have $(p, w) \vdash^a (q, w)$ if $(p, a, q) \in \delta_i$.
– If $a \in \Sigma_c$ and $w \in \Gamma^*$ we have $(p, w) \vdash^a (q, w\gamma)$ if $(p, a, q, \gamma) \in \delta_c$.
– If $a \in \Sigma_r$, $\gamma \in \Gamma \backslash \{\perp\}$ and $w \in \Gamma^*$ we have $(p, w\gamma) \vdash^a (q, w)$ if $(p, a, \gamma, q) \in \delta_r$.
– If $a \in \Sigma_r$ we have $(p, \perp) \vdash^a (q, \perp)$ if $(p, a, \perp, q) \in \delta_r$.

We lift the relation $\vdash$ to words by transitivity and reflexivity, that is, for all $u \in \Sigma^*$, $(q, w) \vdash^{*u} (p, w')$ when the configurations (q, w) and (p, w') are related by a sequence of transitions such that the concatenation of the corresponding labels is the word u. We write $(q, w) \vdash^{\circledast u} (p, w')$ when such a sequence includes a configuration whose state is final. A *trace* of $\mathcal{A}$ on a infinite word $\xi = a_0 a_1 \cdots \in \Sigma^\omega$ is an infinite sequence $(q_0, w_0) \vdash^{a_0} (q_1, w_1) \vdash^{a_1} \cdots$ It is a *final trace* when $q_j \in F$ for infinitely many j's. It is an *accepting trace* when it is a final trace and $(q_0, w_0) = (q_I, \perp)$. The ω-language accepted by $\mathcal{A}$ is $L^\omega(\mathcal{A}) \triangleq \{\xi \in \Sigma^\omega \mid$ there is an accepting trace of $\mathcal{A}$ on $\xi\}$. A language $L \subseteq \Sigma^\omega$ is ω-VPL if $L = L^\omega(\mathcal{A})$ for some VPA $\mathcal{A}$. Two examples of VPA are given at Fig. 1, $\mathcal{A}$ has an accepting trace on $c\,cr\,cr\,cr \ldots$ and so does $\mathcal{B}$ on $crr\,crr \ldots$

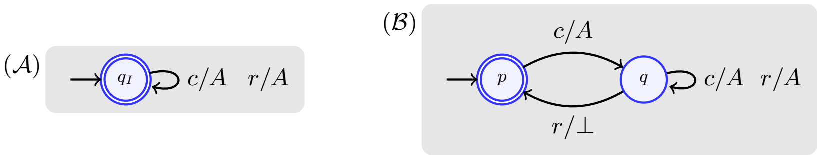

Fig. 1. Two ω-VPA with $\Gamma = \{A, \perp\}$, $\Sigma_i = \emptyset$, $\Sigma_c = \{c\}$ and $\Sigma_r = \{r\}$.

Ultimately Periodic Words. An *ultimately periodic word* is an infinite word $\xi \in \Sigma^\omega$ such that $\xi = uv^\omega$ for some finite *prefix* $u \in \Sigma^*$ and some finite *period* $v \in \Sigma^+$. We call the couple $(u, v) \in \Sigma^* \times \Sigma^+$ a *decomposition* of ξ. Note that ξ admits infinitely many decompositions.

Ultimately periodic words play a central role in our approach as they suffice for the inclusion problem as shown by the following theorem. [4]

Theorem 1. *Let $L, M \subseteq \Sigma^\omega$ be ω-VPL. Then, $L \subseteq M$ iff $\forall uv^\omega \in L, uv^\omega \in M$.*

Matching Relation. The partition of the alphabet $\Sigma = \Sigma_i \cup \Sigma_c \cup \Sigma_r$ induces a unique matching relation between a word's *call* and *return* positions (see [18]). Given $u \in \Sigma^* \cup \Sigma^\omega$ define the *matching relation* of u, denoted $\curvearrowright_u$, as the unique relation on its call and return positions such that for every $j \curvearrowright_u k$ we have $0 \le j < k < |u|$, $u_j \in \Sigma_c$, $u_k \in \Sigma_r$, $|\{n \mid j \curvearrowright_u n\}| \le 1$, $|\{n \mid n \curvearrowright_u k\}| \le 1$ and there are no j', k' with $j' \curvearrowright_u k'$ and $j < j' < k < k'$. Given $j \curvearrowright_u k$ we say that j and k are *matched* positions. A call (resp. return) position j in u is *unmatched*

[4] Theorem 1 can be easily obtained by adapting the proof of Fact 1 in [7].

if $j \curvearrowright_u k$ (resp. $k \curvearrowright_u j$) for no k. Furthermore, for every unmatched positions n in u there is no $j \curvearrowright_u k$ such that $j < n < k$, and if $u_n \in \Sigma_c$ (resp. $u_n \in \Sigma_r$) then there is no unmatched return (resp. call) position k with $n < k$ (resp. $k < n$). A word is said to be *well-matched* if it has no unmatched position.

3 Foundations

In this section we outline our approach which, given a VPA $\mathcal{A} = (Q, q_I, \Gamma, \delta, F)$ and an ω-VPL M, reduces the inclusion problem $L^\omega(\mathcal{A}) \subseteq M$ to finitely many membership queries in M. More precisely, we derive a finite subset S_{finite} of ultimately periodic words of $L^\omega(\mathcal{A})$ such that

$$L^\omega(\mathcal{A}) \subseteq M \iff \forall (u, v) \in S_{\text{finite}},\ uv^\omega \in M \ . \tag{†}$$

Reduction to Legitimate Decompositions. Our first step is to reduce the inclusion check to a subset of ultimately words of $L^\omega(\mathcal{A})$ given by legitimate decompositions. To do so, we define $\mathtt{W}$ as the set of well-matched finite words, $\mathtt{C}$ (resp. $\mathtt{R}$) as the set of finite words where all call (resp. return) positions are matched and $\mathtt{U_c}$ as the set of finite words with at least one unmatched call position. In turn, we define the set of *legitimate decompositions* given by

$$\mathtt{Ld} \triangleq \mathtt{C} \times \mathtt{C} \ \cup \ \mathtt{U_c} \times \mathtt{R}$$

which, as shown next, is sufficient for the inclusion problem between ω-VPL.

Theorem 2. *Let $L, M \subseteq \Sigma^\omega$ be ω-VPL. Then, $L \subseteq M$ iff $\forall (u, v) \in \mathtt{Ld}$, $uv^\omega \in L \implies uv^\omega \in M$.*

Next we leverage the relations $\vdash^*$ and $\vdash^\circledast$ of $\mathcal{A}$ to characterize the legitimate decompositions of the ultimately periodic words of $L^\omega(\mathcal{A})$. We start by defining the following languages of finite words for each pair $p, q \in Q$ of state of $\mathcal{A}$: $L_{p,q} \triangleq \{u \in \Sigma^* \mid \exists w \in \Gamma^*,\ (p, \bot) \vdash^{*u} (q, w)\}$ and $L^\circledast_{p,q} \triangleq \{u \in \Sigma^+ \mid \exists w \in \Gamma^*,\ (p, \bot) \vdash^{\circledast u} (q, w)\}$. Finally, define the following subset of $\mathtt{Ld}$:

$$S \triangleq \bigcup_{p \in Q} L_{q_I, p|\mathtt{C}} \times L^\circledast_{p,p|\mathtt{C}} \ \cup \ L_{q_I, p|\mathtt{U_c}} \times L^\circledast_{p,p|\mathtt{R}}$$

where $L_{|K}$ is defined to be $L \cap K$ to emphasize that L is restricted to K.

Example 1. Consider the VPA $\mathcal{A}$ and $\mathcal{B}$ depicted in Fig. 1. We have $L^\omega(\mathcal{A}) = \mathtt{R}^\omega$, $S = (\mathtt{W} \times \mathtt{W}\backslash\{\epsilon\}) \cup (\mathtt{R}\backslash\mathtt{C} \times \mathtt{R}\backslash\{\epsilon\})$ and $L^\omega(\mathcal{B}) = ((\mathtt{W}\backslash\{\epsilon\})r)^\omega$.

Proposition 1. *We have that $uv^\omega \in L^\omega(\mathcal{A}) \iff \exists (u', v') \in S,\ uv^\omega = u'v'^\omega$.*

By Theorem 2 and Proposition 1 the subset S verifies:

$$L^\omega(\mathcal{A}) \subseteq M \iff \forall (u, v) \in S,\ uv^\omega \in M \ . \tag{1}$$

Next we reduce the inclusion check to a finite subset of S using quasiorders.

Reduction to a Finite Basis. A *quasiorder* (qo) on a set E, is a reflexive and transitive relation $\ltimes \subseteq E \times E$. Given two subsets $X, Y \subseteq E$ the set Y is said to be a *basis* for X with respect to $\ltimes$ whenever $Y \subseteq X$ and $\forall x \in X, \exists y \in Y, y \ltimes x$. A qo $\ltimes$ is a *well-quasiorder* (wqo) if every subset of E admits a finite basis.

We obtain S_{finite} as a finite basis for S with respect to $\leqslant \times \preccurlyeq$ for a pair $\leqslant, \preccurlyeq$ of wqos.[5] To guarantee the direction $\Leftarrow$ in Eq. (†) we need the pair $\leqslant, \preccurlyeq$ to be *M-preserving*, a notion we introduce below.

A pair $\leqslant, \preccurlyeq$ of qos on Σ^* is said to be M-preserving if for all $(u, v), (u', v') \in \mathtt{Ld}$ such that $(u, v), (u', v') \in \mathtt{C} \times \mathtt{C}$ or $(u, v), (u', v') \in \mathtt{U_c} \times \mathtt{R}$,

$$\text{if } uv^\omega \in M, u \leqslant u' \text{ and } v \preccurlyeq v' \text{ then } u'v'^\omega \in M \enspace .$$

Intuitively, M-preservation guarantees that if the inclusion does not hold then the finite basis S_{finite} contains a counterexample.

Next, we fix a pair of M-preserving wqos $\leqslant, \preccurlyeq$ and show the existence of a subset S_{finite} such that Eq. (†) holds. Since $\leqslant \times \preccurlyeq$ is a wqo, there exist two finite bases S_1 and S_2 for $S_{|\mathtt{C}\times\mathtt{C}}$ and $S_{|\mathtt{U_c}\times\mathtt{R}}$ respectively w.r.t. $\leqslant \times \preccurlyeq$. We define S_{finite} to be the union of such sets S_1, S_2, viz. $S_{\text{finite}} \triangleq S_1 \cup S_2 \subseteq S$. We have that: $\forall (u, v) \in S,\ uv^\omega \in M \implies \forall (u, v) \in S_{\text{finite}},\ uv^\omega \in M$. We now turn to the converse implication. Assume that $\forall (u, v) \in S_{\text{finite}},\ uv^\omega \in M$. Let $(u, v) \in S$. If $(u, v) \in S_{|\mathtt{C}\times\mathtt{C}}$ then there is $(u_0, v_0) \in S_1$ such that $(u_0, v_0) \leqslant \times \preccurlyeq (u, v)$. Since $S_1 \subseteq S_{|\mathtt{C}\times\mathtt{C}} \subseteq \mathtt{C} \times \mathtt{C}$ we have that $(u_0, v_0), (u, v) \in \mathtt{C} \times \mathtt{C}$. Since $u_0 v_0^\omega \in M$ and the pair $\leqslant, \preccurlyeq$ is M-preserving, we conclude that $uv^\omega \in M$. The case $(u, v) \in S_{|\mathtt{U_c}\times\mathtt{R}}$ proceeds analogously. It follows that $\forall (u, v) \in S,\ uv^\omega \in M \Longleftarrow \forall (u, v) \in S_{\text{finite}},\ uv^\omega \in M$. Hence, we derive Equation (†) using Equation (1).

In Section 4, we give a fixpoint characterization of S and in Section 5 we show that under some monotonicity conditions on the wqos $\leqslant$ and $\preccurlyeq$ we can effectively compute a finite basis for S. We then give two examples of monotonic pairs of wqos in Section 6. In Section 7 we present our algorithm which given two VPA $\mathcal{A}$ and $\mathcal{B}$ decides the inclusion problem $L^\omega(\mathcal{A}) \subseteq L^\omega(\mathcal{B})$. Therein we discuss the state-based algorithm and give an upper bound on its running time. Finally in Section 8 we report on an empirical evaluation.

4 Fixpoint Characterization

In this section we give a least fixpoint characterization of S for the VPA $\mathcal{A} = (Q, q_I, \Gamma, \delta, F)$. To this end we work with the complete lattice $(\wp(\Sigma^*)^{n\cdot|Q|^2}, \subseteq \times \cdots \times \subseteq)$, where $n \in \{4, 6\}$ and each Cartesian product consists of $n \cdot |Q|^2$ factors.

For a function $f \colon E \to E$ on a quasiordered set $(E, \ltimes)$ and for all $n \in \mathbb{N}$, we define the n-th iterate $f^n : E \to E$ of f inductively as follows: $f^0 \triangleq \lambda x.x$; $f^{n+1} \triangleq f \circ f^n$. The denumerable sequence of *Kleene iterates* of f starting from the bottom value $\bot \in E$ is given by $\{f^n(\bot)\}_{n\in\mathbb{N}}$. Recall that when $(E, \ltimes)$ is a complete lattice and $f \colon E \to E$ is a monotone function (i.e. $d \ltimes d' \implies$

[5] The qo $\leqslant \times \preccurlyeq$ is a wqo when both $\leqslant$ and $\preccurlyeq$ are wqos.

$f(d) \ltimes f(d'))$ then by the Knaster–Tarski theorem, f has a least fixpoint $\operatorname{lfp} f$ given by the supremum of the ascending[6] sequence of Kleene iterates of f.

Given a $n \cdot |Q|^2$-dimensional vector X and a $|Q|^2$-dimensional vector Y on $\wp(\Sigma^*)$ we write $X_{i,p,q}$, for the (i,p,q)-component of X and $Y_{p,q}$ for the (p,q)-component of Y. We define the following equations where $X, X' \in \wp(\mathtt{W})^{|Q|^2}$, $Y, Y' \in \wp(\mathtt{C})^{|Q|^2}$, $Z, Z' \in \wp(\mathtt{R})^{|Q|^2}$, and $T \in \wp(\mathtt{U_c})^{|Q|^2}$:

$$
\begin{aligned}
W(X) &= \langle L_{p,q|(\Sigma_i \cup \{\epsilon\})} \cup \bigcup_{\substack{(p,c,p',\gamma)\in\delta_c,\\ (q',r,\gamma,q)\in\delta_r}} cX_{p',q'}r \cup \bigcup_{q'\in Q} X_{p,q'}\; X_{q',q}\rangle_{p,q\in Q}\\
C(X,Y) &= \langle L_{p,q|\Sigma_r} \cup X_{p,q} \cup \bigcup_{q'\in Q} Y_{p,q'}Y_{q',q}\rangle_{p,q\in Q}\\
R(X,Z) &= \langle L_{p,q|\Sigma_c} \cup X_{p,q} \cup \bigcup_{q'\in Q} Z_{p,q'}Z_{q',q}\rangle_{p,q\in Q}\\
U(Y,Z,T) &= \langle L_{p,q|\Sigma_c} \cup \bigcup_{p',q'\in Q,} Y_{p,p'}T_{p',q'}Z_{q',q}\rangle_{p,q\in Q}\\
W_{\circledast}(X,X') &= \langle L^{\circledast}_{p,q|\Sigma_i} \cup \bigcup_{\substack{(p,c,p',\gamma)\in\delta_c,\\ (q',r,\gamma,q)\in\delta_r,\\ \{p,q\}\cap F\neq\emptyset}} cX_{p',q'}r \cup \bigcup_{\substack{(p,c,p',\gamma)\in\delta_c,\\ (q',r,\gamma,q)\in\delta_r\\ \{p,q\}\cap F=\emptyset}} cX'_{p',q'}r \cup \bigcup_{q'\in Q} (X'_{p,q'}X_{q',q} \cup X_{p,q'}X'_{q',q})\rangle_{p,q\in Q}\\
C_{\circledast}(X',Y,Y') &= \langle L^{\circledast}_{p,q|\Sigma_r} \cup X'_{p,q} \cup \bigcup_{q'\in Q} (Y'_{p,q'}Y_{q',q} \cup Y_{p,q'}Y'_{q',q})\rangle_{p,q\in Q}\\
R_{\circledast}(X',Z,Z') &= \langle L^{\circledast}_{p,q|\Sigma_c} \cup X'_{p,q} \cup \bigcup_{q'\in Q} (Z'_{p,q'}Z_{q',q} \cup Z_{p,q'}Z'_{q',q})\rangle_{p,q\in Q} \; .
\end{aligned}
$$

The equations W, C, R and U are used to obtain the set of words in $\mathtt{W}$, $\mathtt{C}$, $\mathtt{R}$ and $\mathtt{U_c}$ respectively, that connect two configurations of $\mathcal{A}$. The equations $W_{\circledast}$, $C_{\circledast}$ and $R_{\circledast}$ refine those of W, C and R by filtering out words not visiting final states. In turn we define the functions $f_{\mathcal{A}}$ and $r_{\mathcal{A}}$ used to obtain the prefixes u and the periods v respectively for the decompositions $(u,v) \in S$. Define

$$
\begin{aligned}
f_{\mathcal{A}} \colon \wp(\Sigma^*)^{4\cdot|Q|^2} &\longrightarrow \wp(\Sigma^*)^{4\cdot|Q|^2}\\
(X,Y,Z,T) &\longmapsto (W(X), C(X,Y), R(X,Z), U(Y,Z,T))
\end{aligned}
$$

for the prefixes, and for the periods define

$$
\begin{aligned}
r_{\mathcal{A}} \colon \wp(\Sigma^*)^{6\cdot|Q|^2} &\longrightarrow \wp(\Sigma^*)^{6\cdot|Q|^2}\\
(X,Y,Z,X',Y',Z') &\longmapsto \big(W(X),C(X,Y),R(X,Z),W_{\circledast}(X,X'),C_{\circledast}(X',Y,Y'),R_{\circledast}(X',Z,Z')\big) \; .
\end{aligned}
$$

The function $f_{\mathcal{A}}$ (resp. $r_{\mathcal{A}}$) is monotone and the supremum of the ascending sequence of its Kleene iterates starting at the bottom value $\vec{\emptyset} \triangleq (\emptyset, \ldots, \emptyset)$ of dimension $4 \cdot |Q|^2$ (resp. $6 \cdot |Q|^2$) is the vector $(\Lambda_{|\mathtt{W}}, \Lambda_{|\mathtt{C}}, \Lambda_{|\mathtt{R}}, \Lambda_{|\mathtt{U_c}})$ (resp. $(\Lambda_{|\mathtt{W}}, \Lambda_{|\mathtt{C}}, \Lambda_{|\mathtt{R}}, \Lambda^{\circledast}_{|\mathtt{W}}, \Lambda^{\circledast}_{|\mathtt{C}}, \Lambda^{\circledast}_{|\mathtt{R}})$) where $\Lambda_{|\mathtt{J}} \triangleq \langle L_{p,q|\mathtt{J}}\rangle_{p,q\in Q}$ and $\Lambda^{\circledast}_{|\mathtt{J}} \triangleq \langle L^{\circledast}_{p,q|\mathtt{J}}\rangle_{p,q\in Q}$ for $\mathtt{J} \in \{\mathtt{W}, \mathtt{C}, \mathtt{R}, \mathtt{U_c}\}$. Therefore, by the Knaster–Tarski theorem we obtain the following proposition.

Proposition 2. $\operatorname{lfp} f_{\mathcal{A}} = (\Lambda_{|W}, \Lambda_{|C}, \Lambda_{|R}, \Lambda_{|U_c})$ *and* $\operatorname{lfp} r_{\mathcal{A}} = (\Lambda_{|W}, \Lambda_{|C}, \Lambda_{|R}, \Lambda^{\circledast}_{|W}, \Lambda^{\circledast}_{|C}, \Lambda^{\circledast}_{|R})$.

[6] A sequence $\{s_n\}_{n\in\mathbb{N}} \in E^{\mathbb{N}}$ on an ordered set $(E, \ltimes)$ is ascending if for every $n \in \mathbb{N}$ we have $s_n \ltimes s_{n+1}$.

Finally, by Proposition 2, we obtain the desired fixpoint characterization of S:

$$S = \bigcup_{p\in Q}\Big(\big((\mathrm{lfp}\, f_{\mathcal{A}})_{2,q_I,p} \times (\mathrm{lfp}\, r_{\mathcal{A}})_{5,p,p}\big) \cup \big((\mathrm{lfp}\, f_{\mathcal{A}})_{4,q_I,p} \times (\mathrm{lfp}\, r_{\mathcal{A}})_{6,p,p}\big)\Big) \ . \quad (2)$$

Example 2. We derive from the VPA $\mathcal{A}$ depicted in Fig. 1 the following functions

$$\begin{aligned} W(X) &\triangleq \{\epsilon\} \cup cXr \cup XX, & C(X,Y) &\triangleq X \cup YY, \\ R(X,Z) &\triangleq \{c\} \cup X \cup ZZ, & U(Y,Z,T) &\triangleq \{c\} \cup YTZ \ . \end{aligned}$$

Hence, we obtain the function

$$\begin{aligned} f_{\mathcal{A}} \colon \wp(\Sigma^*)^4 &\longrightarrow \wp(\Sigma^*)^4 \\ (X,Y,Z,T) &\longmapsto (W(X), C(X,Y), R(X,Z), U(Y,Z,T)) \ . \end{aligned}$$

The first three iterates of the least fixpoint computation of lfp $f_{\mathcal{A}}$ are given by

$$\begin{aligned} f_{\mathcal{A}}(\vec{\emptyset}) &= (\{\epsilon\}, \emptyset, \{c\}, \{c\}), \\ {f_{\mathcal{A}}}^2(\vec{\emptyset}) &= (\{\epsilon, cr\}, \{\epsilon\}, \{\epsilon, c, c^2\}, \{c\}), \\ {f_{\mathcal{A}}}^3(\vec{\emptyset}) &= (\{\epsilon, cr, c^2r^2, (cr)^2\}, \{\epsilon, cr\}, \{\epsilon, cr, c, c^2, c^3, c^4\}, \{c, c^2, c^3\}) \\ &\ \ \vdots \\ \mathrm{lfp}\, f_{\mathcal{A}} &= (\mathtt{W}, \mathtt{W}, \mathtt{R}, \mathtt{R}\backslash\mathtt{C}) \end{aligned}$$

Since the unique state of $\mathcal{A}$ is a final state we have that $L_{q_I,q_I} = L^{\circledast}_{q_I,q_I}$. Consequently, the function $f_{\mathcal{A}}$ suffices to describe both the set of prefixes and the set of periods of S given by $\big((\mathrm{lfp}\, f_{\mathcal{A}})_2 \times (\mathrm{lfp}\, f_{\mathcal{A}})_2\backslash\{\epsilon\}\big) \cup \big((\mathrm{lfp}\, f_{\mathcal{A}})_4 \times (\mathrm{lfp}\, f_{\mathcal{A}})_3\backslash\{\epsilon\}\big)$.

Each (i,p,q)-component of the Kleene iterates of $f_{\mathcal{A}}$ and $r_{\mathcal{A}}$ keeps a finite set of words. However, if the language $L^{\omega}(\mathcal{A})$ is infinite, the fixpoint computations of lfp $f_{\mathcal{A}}$ and lfp $r_{\mathcal{A}}$ do not terminate in a finite number of steps. Nevertheless, under some monotonicity assumptions on our wqos we show in the following section that we can compute a finite basis for S w.r.t. $\leqslant \times \preccurlyeq$ as a terminating fixpoint computation.

5 Monotonicity Requirements

In order to detect finite bases among the Kleene iterates of the functions defined in the previous section we replace the set inclusion on $\wp(\Sigma^*)$, used so far, with the qo $\sqsubseteq_{\ltimes} \subseteq \wp(\Sigma^*) \times \wp(\Sigma^*)$ defined by $X \sqsubseteq_{\ltimes} Y \overset{\triangle}{\Longleftrightarrow} \forall x \in X, \exists y \in Y, y \ltimes x$. The qo $\sqsubseteq_{\ltimes}$ leverage the notion of basis: given $X \in \wp(\Sigma^*)$ a subset $Y \subseteq X$ is a basis for X with respect to $\ltimes$ whenever $X \sqsubseteq_{\ltimes} Y$.

In the following we lift the notion of basis to n-dimensional vectors component wise and work with the quasiordered sets $(\wp(\Sigma^*)^{n\cdot|Q|^2},\ \sqsubseteq_{\ltimes}^{n\cdot|Q|^2})$, where $n \in \{4,6\}$ and the ordering $\sqsubseteq_{\ltimes}^{n\cdot|Q|^2}$ is given by the product $\sqsubseteq_{\ltimes} \times \cdots \times \sqsubseteq_{\ltimes}$ of $n \cdot |Q|^2$

factors. Given a pair $\leqslant, \preccurlyeq$ of wqos, the orderings $\sqsubseteq_{\leqslant}^{4\cdot|Q|^2}$ and $\sqsubseteq_{\preccurlyeq}^{6\cdot|Q|^2}$ are used to compare the Kleene iterates of the functions $f_{\mathcal{A}}$ and $r_{\mathcal{A}}$ respectively. For them to be apt to detect finite bases for the least fixpoints of these functions the qos $\leqslant$ and $\preccurlyeq$ need to verify some monotonicity conditions.

We introduce the monotonicity conditions $\mathbf{W}, \mathbf{C}, \mathbf{R}, \boldsymbol{C}_{\circledast}, \mathbf{R}_{\circledast}$ and $\mathbf{U}$ on a qo $\ltimes \subseteq \Sigma^* \times \Sigma^*$ as follows: for all $u, u' \in \Sigma^*$ such that $u \ltimes u'$

$$
\begin{array}{lllll}
(\mathbf{W}) & \text{if } u, u' \in \mathtt{W} & \text{and } c \in \Sigma_c, r \in \Sigma_r & \text{then } cur \ltimes cu'r, \\
(\mathbf{C}) & \text{if } u, u' \in \mathtt{C} & \text{and } s \in \mathtt{C}, t \in \Sigma^* & \text{then } sut \ltimes su't, \\
(\mathbf{R}) & \text{if } u, u' \in \mathtt{R} & \text{and } s \in \Sigma^*, t \in \mathtt{R} & \text{then } sut \ltimes su't, \\
(\mathbf{U}) & \text{if } u, u' \in \mathtt{U_c} & \text{and } s \in \mathtt{C}, t \in \mathtt{R} & \text{then } sut \ltimes su't, \\
(\mathbf{C}_{\circledast}) & \text{if } u, u' \in \mathtt{C} & \text{and } s \in \mathtt{C}, t \in \mathtt{C} & \text{then } sut \ltimes su't, \\
(\mathbf{R}_{\circledast}) & \text{if } u, u' \in \mathtt{R} & \text{and } s \in \mathtt{R}, t \in \mathtt{R} & \text{then } sut \ltimes su't.
\end{array}
$$

A pair of qos $\leqslant, \preccurlyeq$ is *monotonic* if $\leqslant$ verifies $\mathbf{W}, \mathbf{C}, \mathbf{R}, \mathbf{U}$ and $\preccurlyeq$ verifies $\mathbf{W}, \mathbf{C}_{\circledast}, \mathbf{R}_{\circledast}$.

Proposition 3. *Let $\leqslant, \preccurlyeq$ be a pair of wqos. There is a positive integer n such that ${f_{\mathcal{A}}}^{n+1}(\vec{\emptyset}) \sqsubseteq_{\leqslant}^{4\cdot|Q|^2} {f_{\mathcal{A}}}^{n}(\vec{\emptyset})$ (resp. ${r_{\mathcal{A}}}^{n+1}(\vec{\emptyset}) \sqsubseteq_{\preccurlyeq}^{6\cdot|Q|^2} {r_{\mathcal{A}}}^{n}(\vec{\emptyset})$); and, if the pair of wqos is monotonic then $\operatorname{lfp} f_{\mathcal{A}} \sqsubseteq_{\leqslant}^{4\cdot|Q|^2} {f_{\mathcal{A}}}^{n}(\vec{\emptyset})$ (resp. $\operatorname{lfp} r_{\mathcal{A}} \sqsubseteq_{\preccurlyeq}^{6\cdot|Q|^2} {r_{\mathcal{A}}}^{n}(\vec{\emptyset})$).*

Each Kleene iterate of $f_{\mathcal{A}}$ and $r_{\mathcal{A}}$ is computable and given a decidable qo $\ltimes$ on Σ^* and two finite sets $X, Y \subseteq \Sigma^*$ it is decidable whether $X \sqsubseteq_{\ltimes} Y$ holds. Thus, given a monotonic pair $\leqslant, \preccurlyeq$ of decidable wqos, by Proposition 3, we can compute a finite basis for $\operatorname{lfp} f_{\mathcal{A}}$ w.r.t. $\leqslant$ and a finite basis for $\operatorname{lfp} r_{\mathcal{A}}$ w.r.t. $\preccurlyeq$. Hence, by Equation (2) we can compute a finite basis for S w.r.t. $\leqslant \times \preccurlyeq$.

6 Quasiorders for ω-VPL

In the following we present two families of qos to solve the inclusion problem $L^\omega(\mathcal{A}) \subseteq M$, the state-based qos which are derived from a VPA-representation of M and compare words according to the set of configurations each word connects in the VPA, and the syntactic qos which rely on the syntactic structure of M. We say that a pair of qos is *M-suitable* if it is an M-preserving and monotonic pair of decidable wqos. Intuitively, if a pair of qos is M-suitable then it can be used in our algorithm to decide the inclusion $L^\omega(\mathcal{A}) \subseteq M$.

State-based Quasiorders. Given a VPA $\mathcal{B} = (\hat{Q}, \hat{q}_I, \hat{\Gamma}, \hat{\delta}, \hat{F})$ we associate with each word $u \in \Sigma^*$ its *context* $\operatorname{ctx}^{\mathcal{B}}[u]$ and *final context* $\operatorname{ctx}_{\circledast}^{\mathcal{B}}[u]$ *in* $\mathcal{B}$ as follows:

$$
\begin{aligned}
\operatorname{ctx}^{\mathcal{B}}[u] &\triangleq \{(p,q) \in \hat{Q}^2 \mid \exists w \in \hat{\Gamma}^*, (p, \bot) \vdash^{*u} (q, w)\}, \\
\operatorname{ctx}_{\circledast}^{\mathcal{B}}[u] &\triangleq \{(p,q) \in \hat{Q}^2 \mid \exists w \in \hat{\Gamma}^*, (p, \bot) \vdash^{\circledast u} (q, w)\} \ .
\end{aligned}
$$

Hence we define the following qos on words in Σ^*:

$$
u \leqslant^{\mathcal{B}} u' \overset{\triangle}{\Longleftrightarrow} \operatorname{ctx}^{\mathcal{B}}[u] \subseteq \operatorname{ctx}^{\mathcal{B}}[u'], \quad u \preccurlyeq^{\mathcal{B}} u' \overset{\triangle}{\Longleftrightarrow} u \leqslant^{\mathcal{B}} u' \wedge \operatorname{ctx}_{\circledast}^{\mathcal{B}}[u] \subseteq \operatorname{ctx}_{\circledast}^{\mathcal{B}}[u'] \ .
$$

Proposition 4. *Let $\mathcal{B}$ be a VPA. The pair $\leqslant^{\mathcal{B}}, \preccurlyeq^{\mathcal{B}}$ is $L^\omega(\mathcal{B})$-suitable.*

Example 3. Consider the pair of qos $\leqslant^{\mathcal{B}}, \preccurlyeq^{\mathcal{B}}$ derived as explained above from $\mathcal{B}$ (Fig. 1) and the set $S = (\mathtt{W} \times \mathtt{W}\backslash\{\epsilon\}) \cup (\mathtt{R}\backslash\mathtt{C} \times \mathtt{R}\backslash\{\epsilon\})$ from Example 1. We have that $\mathrm{ctx}^{\mathcal{B}}[\epsilon] = \{(p,p),(q,q)\}$, $\mathrm{ctx}^{\mathcal{B}}_{\circledast}[\epsilon] = \{(p,p)\}$, $\mathrm{ctx}^{\mathcal{B}}[u] = \{(p,q),(q,q)\}$ and $\mathrm{ctx}^{\mathcal{B}}_{\circledast}[u] = \{(p,q)\}$ for every $u \in \mathtt{R}\backslash\{\epsilon\}$. We have that $\{c\}$ is a basis for $\mathtt{R}\backslash\{\epsilon\}$ w.r.t. $\preccurlyeq^{\mathcal{B}}$ since $c \preccurlyeq^{\mathcal{B}} u$ for every $u \in \mathtt{R}\backslash\{\epsilon\}$. Since $\mathtt{R}\backslash\mathtt{C} \subseteq \mathtt{R}\backslash\{\epsilon\}$ and $\{c\} \subseteq \mathtt{R}\backslash\mathtt{C}$ we deduce that $\{c\}$ is also a basis for $\mathtt{R}\backslash\mathtt{C}$ w.r.t $\leqslant^{\mathcal{B}}$. Similarly we deduce that $\{\epsilon, cr\}$ is basis for $\mathtt{W}$ w.r.t $\leqslant^{\mathcal{B}}$ and that $\{cr\}$ is a basis for $\mathtt{W}\backslash\{\epsilon\}$ w.r.t. $\preccurlyeq^{\mathcal{B}}$. Hence, $(\{\epsilon, cr\} \times \{cr\}) \cup (\{c\} \times \{c\})$ is a basis for S w.r.t. $\leqslant^{\mathcal{B}} \times \preccurlyeq^{\mathcal{B}}$.

Syntactic Quasiorders. Given a ω-VPL M we associate with each word $u \in \Sigma^*$ its *context* $\mathrm{ctx}^M[u]$ and *final context* $\mathrm{ctx}^M_{\circledast}[u]$ *in* M as follows:

$$\begin{aligned}
\mathrm{ctx}^M[u] &\triangleq \{(s,\xi) \in \Sigma^* \times \Sigma^\omega \mid su\xi \in M\},\\
\mathrm{ctx}^M_{\circledast}[u] &\triangleq \{(s,t) \in \Sigma^* \times \Sigma^* \mid s(ut)^\omega \in M\}\ .
\end{aligned}$$

At first glance, we are tempted to define the syntactic qos from ctx^M and $\mathrm{ctx}^M_{\circledast}$ in the analogue way we defined the state-based qos from the contexts and final contexts relatively to a VPA. Although, this definition provides a pair of M-preserving qos, it does not guarantee that the pair is M-suitable. To overcome this, we impose the respect of the partition $\mathcal{P} \triangleq \{\mathtt{W}, \mathtt{C}\backslash\mathtt{W}, \mathtt{R}\backslash\mathtt{W}, \mathtt{U_c}\backslash\mathtt{R}\}$ of Σ^*, meaning that two words compare only if they belong to a same subset of $\mathcal{P}$. Additionally, given $\mathtt{J} \in \mathcal{P}$ we compare two words of $\mathtt{J}$ by considering a restriction of their context and final context in M which depends on $\mathtt{J}$. More precisely, we define the qo $\leqslant^M$ on Σ^* as the union $\bigcup_{\mathtt{J}\in\mathcal{P}} \leqslant^M_{\mathtt{J}}$ where for every $\mathtt{J} \in \mathcal{P}$, the qo $\leqslant^M_{\mathtt{J}} \subseteq \mathtt{J} \times \mathtt{J}$ is defined by

$$\begin{aligned}
u \leqslant^M_{\mathtt{W}} u' &\overset{\triangle}{\iff} \mathrm{ctx}^M[u] \subseteq \mathrm{ctx}^M[u'],\\
u \leqslant^M_{\mathtt{C}\backslash\mathtt{W}} u' &\overset{\triangle}{\iff} \mathrm{ctx}^M[u]_{|\mathtt{C}\times\Sigma^\omega} \subseteq \mathrm{ctx}^M[u']_{|\mathtt{C}\times\Sigma^\omega},\\
u \leqslant^M_{\mathtt{R}\backslash\mathtt{W}} u' &\overset{\triangle}{\iff} \mathrm{ctx}^M[u]_{|\Sigma^*\times\mathtt{R}^\omega} \subseteq \mathrm{ctx}^M[u']_{|\Sigma^*\times\mathtt{R}^\omega},\\
u \leqslant^M_{\mathtt{U_c}\backslash\mathtt{R}} u' &\overset{\triangle}{\iff} \mathrm{ctx}^M[u]_{|\mathtt{C}\times\mathtt{R}^\omega} \subseteq \mathrm{ctx}^M[u']_{|\mathtt{C}\times\mathtt{R}^\omega}\ .
\end{aligned}$$

Similarly, we define the qo $\preccurlyeq^M \triangleq \bigcup_{\mathtt{J}\in\mathcal{P}} \preccurlyeq^M_{\mathtt{J}}$ on Σ^* where for every $\mathtt{J} \in \mathcal{P}$, $\preccurlyeq^M_{\mathtt{J}} \subseteq \mathtt{J} \times \mathtt{J}$ is the qo defined by

$$\begin{aligned}
u \preccurlyeq^M_{\mathtt{W}} u' &\overset{\triangle}{\iff} u \leqslant^M_{\mathtt{W}} u' \ \wedge\ \mathrm{ctx}^M_{\circledast}[u] \subseteq \mathrm{ctx}^M_{\circledast}[u'],\\
u \preccurlyeq^M_{\mathtt{C}\backslash\mathtt{W}} u' &\overset{\triangle}{\iff} u \leqslant^M_{\mathtt{C}\backslash\mathtt{W}} u' \ \wedge\ \left(\mathrm{ctx}^M_{\circledast}[u]_{|\mathtt{C}\times\mathtt{C}} \subseteq \mathrm{ctx}^M_{\circledast}[u']_{|\mathtt{C}\times\mathtt{C}}\right),\\
u \preccurlyeq^M_{\mathtt{R}\backslash\mathtt{W}} u' &\overset{\triangle}{\iff} u \leqslant^M_{\mathtt{R}\backslash\mathtt{W}} u' \ \wedge\ \left(\mathrm{ctx}^M_{\circledast}[u]_{|\Sigma^*\times\mathtt{R}} \subseteq \mathrm{ctx}^M_{\circledast}[u']_{|\Sigma^*\times\mathtt{R}}\right),\\
u \preccurlyeq^M_{\mathtt{U_c}\backslash\mathtt{R}} u' &\overset{\triangle}{\iff} u, u' \in \mathtt{U_c}\backslash\mathtt{R}\ .
\end{aligned}$$

Proposition 5. *Let $\mathcal{B}$ be a VPA. The pair $\leqslant^{L^\omega(\mathcal{B})}, \preccurlyeq^{L^\omega(\mathcal{B})}$ is $L^\omega(\mathcal{B})$-suitable.*

Proof (sketch). First we show that the pair $\leqslant^M, \preccurlyeq^M$ is M-preserving, where $M \triangleq L^\omega(\mathcal{B})$. Let $(u, v), (u', v') \in \mathtt{C} \times \mathtt{C}$ (resp. $\mathtt{U_c} \times \mathtt{R}$) such that $u \leqslant^M u'$, $v \preccurlyeq^M v'$ and $uv^\omega \in M$. From $u \leqslant^M u'$ and $uv^\omega \in M$ we deduce that $(\epsilon, v^\omega) \in \mathrm{ctx}^M_{|\mathtt{C}\times\Sigma^\omega}[u] \subseteq \mathrm{ctx}^M_{|\mathtt{C}\times\Sigma^\omega}[u']$ (resp. $(\epsilon, v^\omega) \in \mathrm{ctx}^M_{|\mathtt{C}\times\mathtt{R}^\omega}[u] \subseteq \mathrm{ctx}^M_{|\mathtt{C}\times\mathtt{R}^\omega}[u']$). Thus, $u'v^\omega \in M$. From $v \preccurlyeq^M v'$ and $u'v^\omega \in M$ we deduce that $(u', \epsilon) \in \mathrm{ctx}^M_{\circledast}[v]_{|\mathtt{C}\times\mathtt{C}} \subseteq \mathrm{ctx}^M_{\circledast\mathtt{C}}[v']_{|\mathtt{C}\times\mathtt{C}}$ (resp. $(u', \epsilon) \in \mathrm{ctx}^M_{\circledast}[v]_{|\Sigma^*\times\mathtt{R}} \subseteq \mathrm{ctx}^M_{\circledast\mathtt{C}}[v']_{|\Sigma^*\times\mathtt{C}}$). Thus, $u'v'^\omega \in M$.

We now show that the qo $\leqslant^M$ satisfies the monotonicity conditions **C** and **R**. Let $u \leqslant^M u'$ such that $u, u' \in \mathtt{C}$ (resp. $u, u' \in \mathtt{R}$). Let $s \in \mathtt{C}$ and $t \in \Sigma^*$ (resp. $s \in \Sigma^*$ and $t \in \mathtt{R}$). If $u, u' \in \mathtt{W}$ then it is easy to check that $sut \leqslant^M su't$. Otherwise $u, u' \in \mathtt{C}\backslash\mathtt{W}$ (resp. $u, u' \in \mathtt{R}\backslash\mathtt{W}$) and we distinguish two cases: if $t \in \mathtt{C}$ (resp. $s \in \mathtt{R}$) then $sut, su't \in \mathtt{C}\backslash\mathtt{W}$ (resp. $sut, su't \in \mathtt{R}\backslash\mathtt{W}$). We show that $sut \leqslant^M_{\mathtt{C}\backslash\mathtt{W}} su't$ (resp. $sut \leqslant^M_{\mathtt{R}\backslash\mathtt{W}} su't$). Let $(s', \xi) \in \mathrm{ctx}^M[sut]_{|\mathtt{C}\times\Sigma^\omega}$ (resp. $(s', \xi) \in \mathrm{ctx}^M[sut]_{|\Sigma^*\times\mathtt{R}^\omega}$). Since $s's \in \mathtt{C}$ (resp. $t\xi \in \mathtt{R}^\omega$), we deduce from $u \leqslant^M_{\mathtt{C}\backslash\mathtt{W}} u'$ (resp. $u \leqslant^M_{\mathtt{R}\backslash\mathtt{W}} u'$) that $(s', \xi) \in \mathrm{ctx}^M[su't]_{|\mathtt{C}\times\Sigma^\omega}$ (resp. $(s', \xi) \in \mathrm{ctx}^M[su't]_{|\Sigma^*\times\mathtt{R}^\omega}$). If $t \in \mathtt{U_c}$ (resp. $s \in \Sigma^*\backslash\mathtt{R}$) then $sut, su't \in \mathtt{U_c}\backslash\mathtt{R}$ and similarly we can show that $sut \leqslant^M_{\mathtt{U_c}\backslash\mathtt{R}} su't$. The proof that $\leqslant^M$ and $\preccurlyeq^M$ are wqos follows from [9, Prop 1.2] by observing that for every $\mathtt{J}$ in the partition $\mathcal{P}$ of Σ^* we have $\leqslant^{\mathcal{B}}_{|\mathtt{J}\times\mathtt{J}} \subseteq \leqslant^M_{\mathtt{J}}$ and $\preccurlyeq^{\mathcal{B}}_{|\mathtt{J}\times\mathtt{J}} \subseteq \preccurlyeq^M_{\mathtt{J}}$, where $\leqslant^{\mathcal{B}}$ and $\preccurlyeq^{\mathcal{B}}$ are the state-based qos previously defined. □

Deciding the syntactic qos can be easily shown to be as hard as the inclusion problem between ω-VPL generated by VPA. Nevertheless, the syntactic qos act as a gold standard for quasiorders in the sense formalized in the next proposition.

Proposition 6. *Let $M \subseteq \Sigma^\omega$ be an ω-VPL and $\leqslant, \preccurlyeq$ be a M-suitable pair of qos such that $\preccurlyeq \subseteq \leqslant$. For every $\mathtt{J} \in \mathcal{P}$ we have $\leqslant_{|J\times J} \subseteq \leqslant^M$ and $\preccurlyeq_{|J\times J} \subseteq \preccurlyeq^M$.*

By Propositions 5 and 6 the pair $\leqslant^{L^\omega(\mathcal{B})}, \preccurlyeq^{L^\omega(\mathcal{B})}$ is the greatest (w.r.t $\subseteq \times \subseteq$) among the $L^\omega(\mathcal{B})$-suitable pairs $\leqslant, \preccurlyeq$ of qos that respect the partition $\mathcal{P}$ and that verify $\preccurlyeq \subseteq \leqslant$.

7 Algorithm

We are now in position to present our algorithm which, given two VPA $\mathcal{A} = (Q, q_I, \Gamma, \delta, F)$ and $\mathcal{B} = (\hat{Q}, \hat{q_I}, \hat{\Gamma}, \hat{\delta}, \hat{F})$ and a pair of $L^\omega(\mathcal{B})$-suitable qos, decides the inclusion problem $L^\omega(\mathcal{A}) \subseteq L^\omega(\mathcal{B})$.

Algorithm 1 computes a finite basis for S w.r.t. $\leqslant \times \preccurlyeq$ (lines 1–2) and afterwards checks membership in $L^\omega(\mathcal{B})$ on every ultimately periodic word uv^ω stemming from this finite basis (lines 3–7).

Theorem 3. *Given the required inputs, Algorithm 1 decides the inclusion problem $L^\omega(\mathcal{A}) \subseteq L^\omega(\mathcal{B})$.*

Proof. As established by Proposition 3, given a monotonic pair $\leqslant, \preccurlyeq$ of decidable wqos, Algorithm 1 computes in line 1 (resp. line 2) a finite basis ${f_{\mathcal{A}}}^m(\vec{\emptyset})$ (resp.

Algorithm 1: Algorithm for deciding $L^\omega(\mathcal{A}) \subseteq L^\omega(\mathcal{B})$

Data: VPA $\mathcal{A} = (Q, q_I, \Gamma, \delta, F)$ and $\mathcal{B} = (\hat{Q}, \hat{q_I}, \hat{\Gamma}, \hat{\delta}, \hat{F})$.
Data: $L^\omega(\mathcal{B})$-suitable pair $\leqslant, \preccurlyeq$.
Data: Procedure deciding $uv^\omega \in L^\omega(\mathcal{B})$ given (u, v).

1 Compute ${f_\mathcal{A}}^m(\vec{\emptyset})$ with least m s.t. ${f_\mathcal{A}}^{m+1}(\vec{\emptyset}) \sqsubseteq_{\leqslant}^{4\cdot|Q|^2} {f_\mathcal{A}}^m(\vec{\emptyset})$;
2 Compute ${r_\mathcal{A}}^{m'}(\vec{\emptyset})$ with least m' s.t. ${r_\mathcal{A}}^{m'+1}(\vec{\emptyset}) \sqsubseteq_{\preccurlyeq}^{6\cdot|Q|^2} {r_\mathcal{A}}^{m'}(\vec{\emptyset})$;
3 **foreach** $p \in Q$ **do**
4   **foreach** $u \in ({f_\mathcal{A}}^m(\vec{\emptyset}))_{2,q_I,p}$, $v \in ({r_\mathcal{A}}^{m'}(\vec{\emptyset}))_{5,p,p}$ **do**
5   **if** $uv^\omega \notin L^\omega(\mathcal{B})$ **then return** false;
6   **foreach** $u \in ({f_\mathcal{A}}^m(\vec{\emptyset}))_{4,q_I,p}$, $v \in ({r_\mathcal{A}}^{m'}(\vec{\emptyset}))_{6,p,p}$ **do**
7   **if** $uv^\omega \notin L^\omega(\mathcal{B})$ **then return** false;
8 **return** true;

${r_\mathcal{A}}^{m'}(\vec{\emptyset}))$ for $\operatorname{lfp} f_\mathcal{A}$ (resp. $\operatorname{lfp} r_\mathcal{A}$) w.r.t. $\leqslant$ (resp. $\preccurlyeq$). Next define:

$$S_\mathcal{A}^{m,m'} \triangleq \bigcup_{p\in Q}\Big((({f_\mathcal{A}}^m(\vec{\emptyset}))_{2,q_I,p} \times ({r_\mathcal{A}}^{m'}(\vec{\emptyset}))_{5,p,p}) \cup (({f_\mathcal{A}}^m(\vec{\emptyset}))_{4,q_I,p} \times ({r_\mathcal{A}}^{m'}(\vec{\emptyset}))_{6,p,p})\Big) \enspace .$$

Using Equation (2) we deduce that $S_\mathcal{A}^{m,m'}$ is a finite basis for S w.r.t. $\leqslant \times \preccurlyeq$. Since the pair $\leqslant, \preccurlyeq$ is $L^\omega(\mathcal{B})$-preserving, by Section 3, we deduce that

$$L^\omega(\mathcal{A}) \subseteq L^\omega(\mathcal{B}) \iff \forall (u,v) \in S_\mathcal{A}^{m,m'},\ uv^\omega \in L^\omega(\mathcal{B}) \enspace .$$

□

We remark that Algorithm 1 can be easily adapted to decide the inclusion problem between visibly pushdown languages of finite words. The adaptation to the finite words case omits the fixpoint computation of line 2 and iterates over the components (i, q_I, p) where $i \in \{2, 3, 4\}$ and where $p \in F$ is a final state.

Example 4. Consider the iterates of the function $f_\mathcal{A}$ from Example 2. One can check that ${f_\mathcal{A}}^4(\vec{\emptyset}) \sqsubseteq^4_{\preccurlyeq^\mathcal{B}} {f_\mathcal{A}}^3(\vec{\emptyset})$ (thus also ${f_\mathcal{A}}^4(\vec{\emptyset}) \sqsubseteq^4_{\leqslant^\mathcal{B}} {f_\mathcal{A}}^3(\vec{\emptyset})$ since $\preccurlyeq^\mathcal{B} \subseteq \leqslant^\mathcal{B}$). Thus, we check whether the inclusion $L^\omega(\mathcal{A}) \subseteq L^\omega(\mathcal{B})$ holds on the finite set $(\{\epsilon, cr\} \times \{cr\}) \cup (\{c, c^2, c^3\} \times \{cr, c, c^2, c^3, c^4\})$ and find the counterexample $c(cr)^\omega \in L^\omega(\mathcal{A}) \backslash L^\omega(\mathcal{B})$.

Antichains Everywhere. We show next that Algorithm 1 remains correct if, in the sequence of Kleene iterates of $f_\mathcal{A}$ or $r_\mathcal{A}$, for each application of $f_\mathcal{A}$ or $r_\mathcal{A}$ we first select a finite basis for their arguments instead (using $\leqslant^{4\cdot|Q|^2}$ for $f_\mathcal{A}$ and $\preccurlyeq^{6\cdot|Q|^2}$ for $r_\mathcal{A}$).

Proposition 7. *Let $\ltimes$ be a qo that verifies the monotonicity conditions* $\mathbf{W}, \mathbf{C}, \mathbf{R}, \mathbf{U}$. *If B is a basis for $(X, Y, Z, T) \in \wp(\mathtt{W})^{|Q|^2} \times \wp(\mathtt{C})^{|Q|^2} \times \wp(\mathtt{R})^{|Q|^2} \times \wp(\mathtt{U_c})^{|Q|^2}$ w.r.t. $\ltimes^{4\cdot|Q|^2}$, then $f_\mathcal{A}(B)$ is a basis for $f_\mathcal{A}(X, Y, Z, T)$ w.r.t. $\ltimes^{4\cdot|Q|^2}$. The analogue result holds for $r_\mathcal{A}$ when $\ltimes$ satisfies the monotonicity conditions* $\mathbf{W}, \mathbf{C}_\circledast, \mathbf{R}_\circledast$.

Since every Kleene iterate of $f_{\mathcal{A}}$ belongs to $\wp(\mathtt{W})^{|Q|^2} \times \wp(\mathtt{C})^{|Q|^2} \times \wp(\mathtt{R})^{|Q|^2} \times \wp(\mathtt{U_c})^{|Q|^2}$ given a basis B for ${f_{\mathcal{A}}}^n(\vec{\emptyset})$ w.r.t. $\leqslant^{4\cdot|Q|^2}$, by Proposition 7, $f_{\mathcal{A}}(B)$ is a basis for ${f_{\mathcal{A}}}^{n+1}(\vec{\emptyset})$ w.r.t. $\leqslant^{4\cdot|Q|^2}$. Hence, at each iteration we can select, for each (i, p, q)-component, a basis w.r.t. $\leqslant$ and then apply $f_{\mathcal{A}}$. In particular, we can keep antichains for each (i, p, q)-component, that is, finite bases of incomparable words. The analogue result holds for the Kleene iterates of $r_{\mathcal{A}}$.

7.1 State-based Algorithm

Next we consider Algorithm 1 instantiated with the pair of state-based qos (§ 6).

Data Structures. Comparing two words given a state-based qo requires to compute the corresponding sets of contexts in $\mathcal{B}$. Instead of computing contexts every time we need to compare two words we cache the context information along with each word for faster retrieval. More precisely, we cache $\mathrm{ctx}^{\mathcal{B}}[u]$ along with u when u is a prefix and we cache $(\mathrm{ctx}^{\mathcal{B}}[v], \mathrm{ctx}^{\mathcal{B}}_{\circledast}[v])$ along with v when v is a period. Next we go even further and explain that new context information can be computed inductively from already computed context information. Assume we are computing a new word during the fixpoint computation, for instance the word cur that is obtained by flanking c and r to u. We will show that the context information of cur can be computed directly from that of u, c and r instead of computing cur from "scratch".

Fixpoint Computation. Given an input vector the functions $f_{\mathcal{A}}$ and $r_{\mathcal{A}}$ add new words of type uu', and cur to its components, where c and r are fixed letters, and u, u' are words already present in some components of the vector. The following equalities show that we can inductively compute the contexts and final contexts in $\mathcal{B}$ of newly added words in these functions: for every $u, u' \in \mathtt{C} \cup \mathtt{R}$, $c \in \Sigma_c$, $r \in \Sigma_r$, we have

$$\mathrm{ctx}^{\mathcal{B}}[uu'] = \{(p,q) \in \hat{Q}^2 \mid \exists p_i \in \hat{Q}, (p, p_i) \in \mathrm{ctx}^{\mathcal{B}}[u], (p_i, q) \in \mathrm{ctx}^{\mathcal{B}}[u']\},$$
$$\mathrm{ctx}^{\mathcal{B}}[cur] = \{(p,q) \in \hat{Q}^2 \mid \exists (p', q') \in \mathrm{ctx}^{\mathcal{B}}[u], \exists \gamma \in \hat{\Gamma}, (p, c, p', \gamma) \in \hat{\delta}_c, (q', r, \gamma, q) \in \hat{\delta}_r\} \ .$$

The definitions for $\mathrm{ctx}^{\mathcal{B}}_{\circledast}[uu']$ and $\mathrm{ctx}^{\mathcal{B}}_{\circledast}[cur]$ are left as exercise to the reader.

Example 5. Using the above definition it is routine to check that $\mathrm{ctx}^{\mathcal{B}}[c\,r] = \{(p,q),(q,q)\}$ because $cr = c\epsilon r$, $\mathrm{ctx}^{\mathcal{B}}[\epsilon] = \{(p,p),(q,q)\}$ (Example 3) and $(p,c,q,A),(q,c,q,A) \in \hat{\delta}_c$, $(q,r,A,q) \in \hat{\delta}_r$.

Using the context information cached along words we check convergence of the fixpoint computations (lines 1–2) using the following qos directly on contexts $\sqsubseteq_{\subseteq}$ on $\wp(\wp(\hat{Q}^2))^4$ for prefixes and $\sqsubseteq_{\subseteq\times\subseteq}$ on $\wp(\wp(\hat{Q}^2) \times \wp(\hat{Q}^2))^6$ for periods.

Incidentally, as we show below, we can perform the membership checks of lines 5 and 7 (asking whether $uv^\omega \in L^\omega(\mathcal{B})$ given u and v) using the context information associated to the prefix u and period v and nothing else.

Membership Check. To decide membership in $L^\omega(\mathcal{B})$ we use the membership predicate $\mathrm{Inc}^{\mathcal{B}}$ defined for $x, y_1, y_2 \in \wp(\hat{Q}^2)$ as follows:

$$\mathrm{Inc}^{\mathcal{B}}(x, y_1, y_2) \triangleq \exists q, p \in \hat{Q}, (\hat{q_I}, q) \in x \wedge (q, p) \in y_1^* \wedge (p, p) \in y_1^* \circ y_2 \circ y_1^* \ ,$$

where, given two binary relations $y, y' \in \wp(\hat{Q}^2)$ on states of $\mathcal{B}$, the notation $y \circ y'$ denotes their composition, and y^* denotes the Kleene closure of y.

Proposition 8. *For all* $(u, v) \in \mathtt{Ld}$*,* $\mathit{Inc}^{\mathcal{B}}(\mathit{ctx}^{\mathcal{B}}[u], \mathit{ctx}^{\mathcal{B}}[v], \mathit{ctx}_{\circledast}^{\mathcal{B}}[v]) \iff uv^\omega \in L^\omega(\mathcal{B})$.

Proof. Let $(u, v) \in \mathtt{Ld}$. Note that if $v \in \mathtt{C}$ (resp. $v \in \mathtt{R}$) then for every positive integer n we have $v^n \in \mathtt{C}$ (resp. $v^n \in \mathtt{R}$) and $(p, q) \in \mathrm{ctx}^{\mathcal{B}}[v]^* \iff \exists n, (p, q) \in \mathrm{ctx}^{\mathcal{B}}[v^n]$. Therefore, if $\mathrm{Inc}^{\mathcal{B}}(\mathrm{ctx}^{\mathcal{B}}[u], \mathrm{ctx}^{\mathcal{B}}[v], \mathrm{ctx}_{\circledast}^{\mathcal{B}}[v])$ holds then there are $q, p \in \hat{Q}$ and two positive integers n, m such that $(\hat{q_I}, q) \in \mathrm{ctx}^{\mathcal{B}}[u]$, $(q, p) \in \mathrm{ctx}^{\mathcal{B}}[v^n]$ and $(p, p) \in \mathrm{ctx}_{\circledast}^{\mathcal{B}}[v^m]$. If $(u, v) \in \mathtt{C} \times \mathtt{C}$ then we deduce an accepting trace of $\mathcal{B}$ on uv^ω of the form $(\hat{q_I}, \bot) \vdash^{*u} (q, \bot) \vdash^{*v^n} (p, \bot) \vdash^{\circledast v^m} (p, \bot)$ for uv^ω. If $(u, v) \in \mathtt{U_c} \times \mathtt{R}$ then we deduce an accepting trace of $\mathcal{B}$ on uv^ω of the form $(\hat{q_I}, \bot) \vdash^{*u} (q, w) \vdash^{*v^n} (p, ww') \vdash^{\circledast v^m} (p, ww'w'')$ for some $w, w', w'' \in \Gamma$.

Conversely if $uv^\omega \in L^\omega(\mathcal{B})$ then there is an accepting trace of $\mathcal{B}$ on uv^ω.

- If $(u, v) \in \mathtt{C} \times \mathtt{C}$ then this trace is of the form

$$(\hat{q_I}, \bot) \vdash^{*u} (q, \bot) \vdash^{*v} (q_1, \bot) \vdash^{*v} (q_2, \bot) \vdash^{*v} \cdots$$

  Since $\hat{Q}$ is finite, there is $p \in \hat{Q}$ and a sequence $\{n_k\}_{k \in \mathbb{N}}$ such that $q_{n_k} = p$ for all $k \in \mathbb{N}$. Since the trace is accepting there is $m \in \mathbb{N}$ such that $(p, \bot) \vdash^{\circledast v^m} (p, \bot)$.
- If $(u, v) \in \mathtt{U_c} \times \mathtt{R}$ then it is of the form

$$(\hat{q_I}, \bot) \vdash^{*u} (q, w_0) \vdash^{*v} (q_1, w_1) \vdash^{*v} (q_2, w_1 w_2) \vdash^{*v} \cdots$$

  where for each $j \in \mathbb{N}$ no symbol of w_j is popped while reading v in the sequence of transitions $(q_j, w_j) \vdash^{*v} (q_{j+1}, w_j w_{j+1})$. Thus, we can derive sequences $(q_j, \bot) \vdash^{*v} (q_{j+1}, w_{j+1})$ for every $j \in \mathbb{N}$. There is $p \in \hat{Q}$ and a sequence $\{n_k\}_{k \in \mathbb{N}}$ such that $q_{n_k} = p$ for all $k \in \mathbb{N}$ and since the trace is accepting there is $m \in \mathbb{N}$ such that $(p, \bot) \vdash^{\circledast v^m} (p, w_{n_j} \cdots w_{n_{j+m}})$.

In both cases we deduce that $(\hat{q_I}, q) \in \mathrm{ctx}^{\mathcal{B}}[u]$, $(q, p) \in \mathrm{ctx}^{\mathcal{B}}[v^{n_0}]$ and $(p, p) \in \mathrm{ctx}_{\circledast}^{\mathcal{B}}[v^m]$. Thus, $\mathrm{Inc}^{\mathcal{B}}(\mathrm{ctx}^{\mathcal{B}}[u], \mathrm{ctx}^{\mathcal{B}}[v], \mathrm{ctx}_{\circledast}^{\mathcal{B}}[v])$ holds. $\square$

By showing how to reason on contexts directly (for comparisons, for applying functions $f_{\mathcal{A}}$ and $r_{\mathcal{A}}$, for convergence check and for membership check) we removed the need to store words altogether since their contexts suffice. To sum up, Algorithm 1 instantiated with the state-based qos can be implemented by manipulating directly subsets of $\wp(\hat{Q}^2)$ (for the prefixes) and pairs of subsets of $\wp(\hat{Q}^2)$ (for the periods) thereby removing the need to store and manipulate words. We call this implementation of Algorithm 1 the *state-based algorithm*. We conclude this section with its complexity.

Proposition 9. *Let* $n \triangleq |Q|$*,* $\hat{n} \triangleq |\hat{Q}|$ *and* $m \triangleq \mathit{max}\{1, |\Sigma|\}$*. The running time of the state-based algorithm is* $2^{O(\hat{n}^2)} m^2 n^4$.

8 Experiments

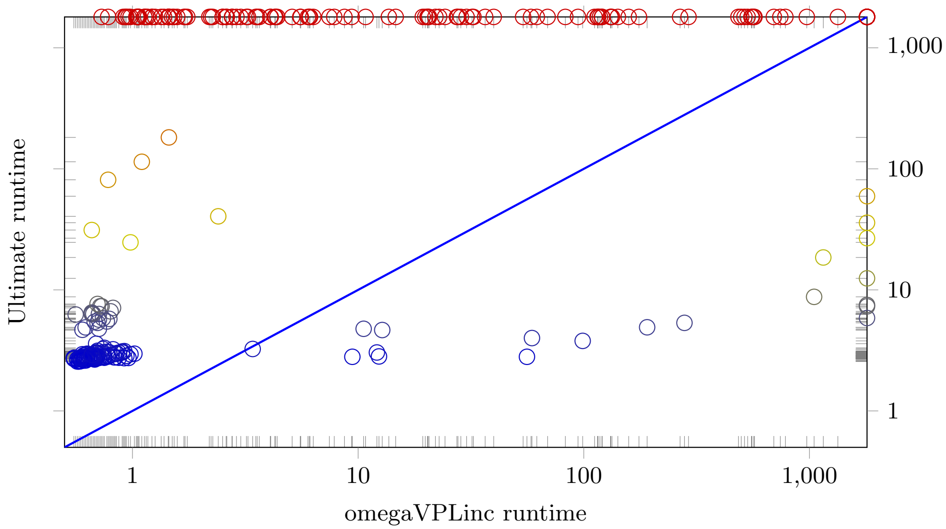

Fig. 2. Scatter plot comparing the runtime (in seconds) of Ultimate and omegaVPLinc on the Ultimate suite. Both axis feature a logarithmic scale. When a tool does not return an answer within 1800 seconds (it runs out of time or memory) the data point is plotted on the edge thereof (top edge for Ultimate, right edge for omegaVPLinc).

We implemented *omegaVPLinc* [11] , a Java prototype of the state-based algorithm and evaluated it against Ultimate from Heizmann et al. [21] which decides inclusion via complementation, intersection and emptiness check.[7]

Benchmarks. Our experiments use two sets of benchmarks. The first stems from [18] and consists of 5 queries $L^\omega(\mathcal{A}) \subseteq L^\omega(\mathcal{B})$ given $\mathcal{A}$ and $\mathcal{B}$. We first translated those VPA into the AutomataScript language that Ultimate and omegaVPLinc can use and then we minimized them with Ultimate. The second set of benchmarks consists of 281 instances of VPA $\mathcal{A}, \mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n$ for which we run the query $L^\omega(\mathcal{A}) \subseteq \bigcup_{i=1}^{n} L^\omega(\mathcal{B}_i)$. These VPA were computed by Ultimate from randomly selected tasks in SV-COMP (Software Verification Competition) termination category. We used Ultimate to compute the unions of $\mathcal{B}_1, \ldots, \mathcal{B}_n$ and then minimize the result before running each query.

[7] We excluded FADecider [18] from our evaluation because it returned 22 false positive answers on a randomly chosen subset of 50 from our 286 benchmarks. Counterexamples to inclusion for these benchmarks were validated with Ultimate. The problem has been reported.

Experimental Setup. We ran our experiment in Debian/GNU Linux 11 (Bullseye) 64bit, running on a server with 20 GB of RAM and 2 Xeon E5640 2.6 GHz CPUs. We used Ultimate version 0.2.1, with openJDK 11.0.13, whereas omegaVPLinc uses openJDK 17.0.1. Maximal heap size for both programs was set to 6 GB and they were given a timeout of 30 minutes (or, equivalently, 1800 seconds).

Results. Of the 5 benchmarks in the FADecider suite, omegaVPLinc is faster on 4 of them. Our prototype times out on the remaining one, while Ultimate runs out of memory. Of the 281 benchmarks in the Ultimate suite, omegaVPLinc correctly returns an answer on 253 (165 $\subseteq$ and 88 $\nsubseteq$), times out on 27 and runs out of memory on 1. Ultimate, however, only terminates on 142 benchmarks, running out of memory on the remaining 139 (the red data points on the top edge in Fig. 2). There are 7 benchmarks for which Ultimate terminates, but omegaVPLinc doesn't (the data points on the right edge but not the top one), whereas there are 118 benchmarks for which omegaVPLinc terminates, but Ultimate doesn't (the red data points on the top edge but not the right one). Of the 135 benchmarks on which both tools terminate, omegaVPLinc is faster than Ultimate on 123 (data points touching no edges and above the diagonal). Moreover omegaVPLinc and Ultimate coincide on whether inclusion holds (98) or not (37). This empirical evaluation suggests that omegaVPLinc scales up better than Ultimate on both of these benchmark sets.

9 Conclusion and Future Work

We presented novel algorithms to solve the inclusion problem between visibly pushdown languages of infinite words that leverage antichain-like techniques as well as the use of separate quasiorders for prefixes and periods of ultimately periodic words. Our empirical evaluation suggests that our approach scales up better than the ones relying on an explicit complementation. A future work is to extend our approach to the class of operator-precedence languages [15] which also enjoy an EXPTime-complete inclusion problem and which is strictly contained in the class of deterministic CFL, and strictly contains VPL [8].

References

1. Abdulla, P.A., Chen, Y.F., Clemente, L., Holík, L., Hong, C.D., Mayr, R., Vojnar, T.: Simulation Subsumption in Ramsey-Based Büchi Automata Universality and Inclusion Testing. In: CAV'10: Proc. 20th Int. Conf. on Computer Aided Verification. Springer (2010). https://doi.org/10.1007/978-3-642-14295-6_14
2. Abdulla, P.A., Chen, Y.F., Clemente, L., Holík, L., Hong, C.D., Mayr, R., Vojnar, T.: Advanced Ramsey-Based Büchi Automata Inclusion Testing. In: CONCUR'11: Proc. 22nd Int. Conf. on Concurrency Theory. Springer (2011). https://doi.org/10.1007/978-3-642-23217-6_13
3. Abdulla, P.A., Chen, Y.F., Holík, L., Mayr, R., Vojnar, T.: When Simulation Meets Antichains. In: TACAS'10: Proc. 16th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems. Springer (2010). https://doi.org/10.1007/978-3-642-12002-2_14

4. Alur, R., Madhusudan, P.: Visibly pushdown languages. In: STOC'04: Proc. 36th Ann. ACM Symp. on Theory of Computing. ACM (2004). https://doi.org/10.1145/1007352.1007390
5. Bouajjani, A., Habermehl, P., Holík, L., Touili, T., Vojnar, T.: Antichain-Based Universality and Inclusion Testing over Nondeterministic Finite Tree Automata. In: CIAA'08: Proc. Int. Conf. on Implementation and Applications of Automata. LNCS, Springer (2008). https://doi.org/10.1007/978-3-540-70844-5_7
6. Bruyère, V., Ducobu, M., Gauwin, O.: Visibly Pushdown Automata: Universality and Inclusion via Antichains. In: LATA'13: Proc. Int. Conf. on Language and Automata Theory and Applications. LNCS, Springer (2013). https://doi.org/10.1007/978-3-642-37064-9_18
7. Calbrix, H., Nivat, M., Podelski, A.: Ultimately periodic words of rational ω-languages. In: Proc. Int. Symp. on Mathematical Foundations of Programming Semantics (MFPS). LNCS, Springer (1994). https://doi.org/10.1007/3-540-58027-1_27
8. Crespi Reghizzi, S., Mandrioli, D.: Operator precedence and the visibly pushdown property. Journal of Computer and System Sciences **78**(6), 1837–1867 (2012). https://doi.org/10.1016/j.jcss.2011.12.006
9. de Luca, A., Varricchio, S.: Well quasi-orders and regular languages. Acta Informatica **31**(6), 539–557 (1994). https://doi.org/10.1007/BF01213206
10. De Wulf, M., Doyen, L., Henzinger, T.A., Raskin, J.F.: Antichains: A new algorithm for checking universality of finite automata. In: CAV'06: Proc. 16th Int. Conf. on Computer Aided Verification. Springer (2006). https://doi.org/10.1007/11817963_5
11. Doveri, K., Ganty, P., Hadzi-Djokic, L.: omegavplinc v1.1 (Jan 2023). https://doi.org/10.5281/zenodo.7506895
12. Doveri, K., Ganty, P., Mazzocchi, N.: FORQ-Based Language Inclusion Formal Testing. In: CAV'22: Proc. 32nd Int. Conf. on Computer Aided Verification. Springer (2022). https://doi.org/10.1007/978-3-031-13188-2_6
13. Doveri, K., Ganty, P., Parolini, F., Ranzato, F.: Inclusion Testing of Büchi Automata Based on Well-Quasiorders. In: CONCUR'21: Proc. 32nd Int. Conf. on Concurrency Theory. LIPIcs, Schloss Dagstuhl (2021). https://doi.org/10.4230/LIPIcs.CONCUR.2021.3
14. Doyen, L., Raskin, J.F.: Antichain Algorithms for Finite Automata. In: Tools and Algorithms for the Construction and Analysis of Systems. LNCS, Springer (2010)
15. Floyd, R.W.: Syntactic analysis and operator precedence. J. ACM **10**(3), 316–333 (1963). https://doi.org/10.1145/321172.321179
16. Fogarty, S., Vardi, M.Y.: Efficient Büchi Universality Checking. In: TACAS'10: Proc. 16th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems. LNCS, Springer (2010). https://doi.org/10.1007/978-3-642-12002-2_17
17. Friedmann, O., Klaedtke, F., Lange, M.: Ramsey goes visibly pushdown. In: ICALP'13: Proc. 40th Int. Coll. on Automata, Languages, and Programming. LNCS, Springer (2013). https://doi.org/10.1007/978-3-642-39212-2_22
18. Friedmann, O., Klaedtke, F., Lange, M.: Ramsey-based inclusion checking for visibly pushdown automata. ACM Transactions on Computational Logic **16**(4), 1–24 (2015). https://doi.org/10.1145/2774221
19. Ganty, P., Ranzato, F., Valero, P.: Language inclusion algorithms as complete abstract interpretations. In: Static Analysis. Springer (2019). https://doi.org/10.1007/978-3-030-32304-2_8

20. Ganty, P., Ranzato, F., Valero, P.: Complete abstractions for checking language inclusion. ACM Trans. Comput. Logic **22**(4) (2021). https://doi.org/10.1145/3462673
21. Heizmann, M., Chen, Y., Dietsch, D., Greitschus, M., Hoenicke, J., Li, Y., Nutz, A., Musa, B., Schilling, C., Schindler, T., Podelski, A.: Ultimate automizer and the search for perfect interpolants - (competition contribution). In: TACAS'18: Proc. 24th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems. LNCS, Springer (2018). https://doi.org/10.1007/978-3-319-89963-3_30
22. Heizmann, M., Hoenicke, J., Podelski, A.: Nested interpolants. SIGPLAN Notices **45**(1), 471–482 (2010). https://doi.org/10.1145/1707801.1706353
23. Hieronymi, P., Ma, D., Oei, R., Schaeffer, L., Schulz, C., Shallit, J.: Decidability for Sturmian Words. In: Manea, F., Simpson, A. (eds.) CSL'22: Proc. 30th EACSL Ann. Conf. on Computer Science Logic. LIPIcs, Schloss Dagstuhl (2022). https://doi.org/10.4230/LIPIcs.CSL.2022.24
24. Meyer, R., Muskalla, S., Neumann, E.: Liveness verification and synthesis: New algorithms for recursive programs. CoRR **abs/1701.02947** (2017), http://arxiv.org/abs/1701.02947
25. Oei, R., Ma, D., Schulz, C., Hieronymi, P.: Pecan: An automated theorem prover for automatic sequences using Büchi automata (2021), https://arxiv.org/abs/2102.01727
26. Picalausa, F., Servais, F., Zimányi, E.: Xevolve: An XML schema evolution framework. In: SAC'11: Proc. ACM Symp. on Applied Computing. p. 1645–1650. ACM (2011). https://doi.org/10.1145/1982185.1982530
27. Rajasekaran, A., Shallit, J., Smith, T.: Additive number theory via automata theory. Theory of Computing Systems **64**(3), 542–567 (2019). https://doi.org/10.1007/s00224-019-09929-9