# The height of piecewise-testable languages with applications in logical complexity*

**Prateek Karandikar**[1] **and Philippe Schnoebelen**[2]

1 **IRIF, Université Paris Diderot & LIPN, Université Paris 13, France**
2 **LSV, CNRS & ENS Cachan, Université Paris-Saclay, France**

**Abstract**

The height of a piecewise-testable language $L$ is the maximum length of the words needed to define $L$ by excluding and requiring given subwords. The height of $L$ is an important descriptive complexity measure that has not yet been investigated in a systematic way. This paper develops a series of new techniques for bounding the height of finite languages and of languages obtained by taking closures by subwords, superwords and related operations.

As an application of these results, we show that $\mathsf{FO}^2(A^*, \sqsubseteq)$, the two-variable fragment of the first-order logic of sequences with the subword ordering, can only express piecewise-testable properties and has elementary complexity.

**1998 ACM Subject Classification** F.4.1 Mathematical Logic, F.4.3 Formal Languages, F.3.1 Specifying and Verifying and Reasoning about Programs

**Digital Object Identifier** 10.4230/LIPIcs.CSL.2016.32

## 1 Introduction

For two words $u$ and $v$ and some $n \in \mathbb{N}$, we write $u \sim_n v$ when $u$ and $v$ have the same (scattered) subwords[1] of length at most $n$. A language $L \subseteq A^*$ is *$n$-piecewise-testable* (or just "$n$-PT") if it is closed under $\sim_n$, or, equivalently, if it can be obtained as a boolean combination of *principal filters* of the form $A^*a_1A^*a_2A^* \cdots a_\ell A^*$ with $\ell \leqslant n$, and where $a_1, \ldots, a_\ell$ are letters from $A$. For example, with $A = \{a, b, c\}$, the language $a^+b^*$ is 2-PT since it can be obtained as $A^*aA^* \cap \neg A^*cA^* \cap \neg A^*bA^*aA^*$. Thus $a^+b^*$ can be described as "all words that have $a$ but neither $c$ nor $ba$ as a subword". Finally, we say that $L$ is piecewise-testable if it is $n$-piecewise-testable for some $n$ and the smallest such $n$ is called the piecewise-testability *height* of $L$, denoted $h(L)$ in this paper. We write $\mathsf{PT}$ for the class of piecewise-testable languages (over some alphabet $A$) and $\mathsf{PT}_n$ for the languages with height at most $n$, so that $\mathsf{PT}_0 \subseteq \mathsf{PT}_1 \subseteq \cdots \mathsf{PT}_n \subseteq \cdots \mathsf{PT}$ form a cumulative hierarchy of varieties of regular languages.

Piecewise-testable (PT) languages were introduced more than forty years ago in Simon's doctoral thesis (see [29, 27]) and have played an important role in the algebraic and logical theory of first-order definable languages, see [25, 4, 15] and the references therein. They also constitute an important class of simple regular languages with applications in learning theory [17], databases [2], linguistics [26], etc. The concept of PT languages has been extended to encompass trees [2], infinite words [24], pictures [23], etc.

The height of piecewise-testable languages is a natural measure of descriptive complexity. Indeed, $h(L)$ coincides with the number of variables needed in a $\mathcal{B}\Sigma_1$ formula that defines

* This work was partially supported by ANR grant ANR-14-CE28-0002 PACS.
[1] Or "subsequences", not to be confused with "factors".


$L$ [4]. In this paper, the main question we address is "*how can one bound the height of PT languages obtained by natural language-theoretic operations?*" Since the height of these languages is a more robust measure than, say, their state complexity, it can be used advantageously in the complexity analysis of problems where PT languages are prominent. As a matter of fact, our results apply to open problems in the logic of subwords, see section 7.

**Related work.** The height of PT languages has been used to measure the difference between separable languages, see e.g. [9]. However the literature is quite poor on the question of estimating the height of PT languages. Algorithms that decide whether a regular language $L$ (given e.g. by its canonical DFA $\mathcal{A}_L$) is PT usually provide a bound on $h(L)$ in terms of $\mathcal{A}_L$: recently Klíma and Polák showed that $h(L)$ is bounded by the depth of $\mathcal{A}_L$ [16]. The currently best bounds on $h(L)$ based on automata for $L$ have been obtained by Masopust and Thomazo [22, 21].

When $L$ is obtained by operations on other languages, very little is known about PT heights. It is clear that $h(A^* \setminus L) = h(L)$ and that $h(L \cup L') \leqslant \max(h(L), h(L'))$ but beyond boolean operations, quotients, and inverse morphisms, there are very few known ways of obtaining PT languages (see Appendix A).

**Our contribution.** We provide upper bounds on the PT height of finite languages and on PT-languages obtained by downward-closure (collecting all subwords of all words from some $L$), upward-closure, and some related operations (collecting words in $L$ that are minimal wrt the subword ordering, etc.) We also show that the incomparability relation preserves piecewise-testability and bound the PT heights of the resulting languages. Crucially, we show that these bounds are *polynomial* when expressed in terms of the PT height of the arguments. One important tool is a *small-subword theorem* that shows how any long word $u$ contains a short subword $u'$ that is $\sim_n$-equivalent. Reasoning about subwords involves ad hoc techniques and leveraging the small-subword theorem to analyse downward-closures or incomparability languages turns out to be non-trivial. Subsequently, all the above results are used to prove that $\mathsf{FO}^2(A^*, \sqsubseteq)$, the two-variable logic of subwords, has elementary complexity. For this logic, the decidability proof in [14] did not come with an elementary complexity upper bound because the usual measures of complexity for regular languages can grow exponentially with each Boolean combination of upward and downward closures, and this is what prompted our investigation of PT heights.

**Outline of the paper.** Section 2 recalls the basic notions (subwords, $\sim_n$, ..) and gives some first bounds relating PT heights and canonical automata. Section 3 focuses on finite languages and develops our main tool: the small-subword theorem. Sections 4 and 5 give bounds for the height of PT languages obtained by upward and downward closures, while Section 6 considers the incomparability relation and the resulting PT heights. Finally, in Section 7 we apply these results to the complexity of $\mathsf{FO}^2(A^*, \sqsubseteq)$. Several proofs have been relegated to the Appendix, usually when the underlying techniques are not used in later developments.

## 2 Basic notions

We consider finite words $u, v, \ldots$ over a given finite alphabet $A$ of letters like $a, b, \ldots$. Concatenation of words is written multiplicatively, with the empty word $\varepsilon$ as unit. We freely use regular expressions like $(ab)^* + (ba)^*$ to denote regular languages.

The length of a word $u$ is written $|u|$ while, for a letter $a \in A$, $|u|_a$ denotes the number of occurrences of $a$ in $u$. The set of all words over $A$ is written $A^*$ and for $\ell \in \mathbb{N}$ we use $A^{=\ell}$ and $A^{\leqslant \ell}$ to denote the subsets of all words of length $\ell$ and of length at most $\ell$ respectively.

A word $v$ is a *factor* of $u$ if there exist words $u_1$ and $u_2$ such that $u = u_1 v u_2$. If furthermore $u_1 = \varepsilon$ then $v$ is a *prefix* of $u$ and we write $v^{-1}u$ to denote the residual $u_2$. If $u_2 = \varepsilon$ then $v$ is a *suffix* and $u\,v^{-1}$ is the residual.

**Subwords.** We say that a word $u$ is a *subword* (i.e., a subsequence) of $v$, written $u \sqsubseteq v$, when $u$ is some $a_1 \cdots a_n$ and $v$ can be written as $v_0 a_1 v_1 \cdots a_n v_n$ for some $v_0, v_1, \ldots, v_n \in A^*$, e.g., $\varepsilon \sqsubseteq bba \sqsubseteq ababa$. We write $u \sqsubset v$ for the associated strict ordering, where $u \neq v$. Two words $u$ and $v$ are *incomparable* (with respect to the subword relation), denoted $u \perp v$, if $u \not\sqsubseteq v$ and $v \not\sqsubseteq u$. Factors are a special case of subwords.

With any $u \in A^*$ we associate the upward and downward closures, $\uparrow u$ and $\downarrow u$, given by[2]

$$\uparrow u \stackrel{\text{def}}{=} \{v \in A^* \mid u \sqsubseteq v\}\,, \qquad \downarrow u \stackrel{\text{def}}{=} \{v \in A^* \mid v \sqsubseteq u\}\,.$$

For example, $\downarrow ab = \{ab, a, b, \varepsilon\}$ and $\uparrow ab = A^*aA^*bA^*$. We also consider the strict superwords and subwords, with $\uparrow_< u \stackrel{\text{def}}{=} \{v \mid u \sqsubset v\}$ and $\downarrow_< u \stackrel{\text{def}}{=} \{v \mid v \sqsubset u\}$. This is generalised to the closures of whole languages, via e.g. $\uparrow L = \bigcup_{u \in L} \uparrow u$ and $\downarrow_< L = \bigcup_{u \in L} \downarrow_< u$. We say that a language $L$ is *upward-closed* if $L = \uparrow L$, and *downward-closed* if $L = \downarrow L$. Note that a language is upward-closed if, and only if, its complement is downward-closed. It is known that upward-closed and downward-closed languages are regular (Haines Theorem [6], also a corollary of Higman's Lemma [8]) so $\uparrow L$, $\downarrow L$, $\uparrow_< L$ and $\downarrow_< L$ are regular for any $L$. Finally we further define

$$I(L) \stackrel{\text{def}}{=} \{u \in A^* \mid \exists v \in L : u \perp v\}\,.$$

Thus $I(L)$ collects all words that are incomparable with *some word* in $L$.

**Simon's congruence and piecewise-testable languages.** For $n \in \mathbb{N}$ and $u, v \in A^*$, we let

$$u \sim_n v \iff \downarrow u \cap A^{\leqslant n} = \downarrow v \cap A^{\leqslant n}\,. \qquad u \lesssim_n v \iff u \sim_n v \wedge u \sqsubseteq v\,. \tag{1}$$

Note that $\lesssim_n$ is stronger than $\sim_n$. Both relations are (pre)congruences: $u \sim_n v$ and $u' \sim_n v'$ imply $uu' \sim_n vv'$, while $u \lesssim_n v$ and $u' \lesssim_n v'$ imply $uu' \lesssim_n vv'$. The equivalence $\sim_n$ is called Simon's congruence of order $n$. We write $[u]_n$ for the equivalence class of $u \in A^*$ under $\sim_n$. Note that each $\sim_n$, for $n = 1, 2, \ldots$, has finite index.

There exist several characterisations of piecewise-testable languages: in the introduction we said that $L \subseteq A^*$ is $n$-piecewise-testable (or "$n$-PT") if it is a boolean combination of principal filters $A^*a_1A^*a_2A^* \cdots a_\ell A^*$ (i.e., of closures $\uparrow a_1 a_2 \cdots a_\ell$) with $\ell \leqslant n$. Equivalently, $L$ is $n$-PT if it is a union $[u_1]_n \cup \cdots \cup [u_m]_n$ of $\sim_n$-classes. The first definition is convenient when we want to show that $L$ is $n$-PT: we describe it in terms of required and excluded subwords and check the length of these subwords, as when we showed that $a^+b^*$ is 2-PT. The second characterisation is convenient when we want to show that $L$ is not $n$-PT: by exhibiting two words $u \sim_n v$ such that $u \in L$ and $v \notin L$, one proves that $L$ is not saturated by $\sim_n$. E.g., $a^+b^*$ is not 1-PT since $ab \sim_1 ba$ while only $ab$ is in $a^+b^*$.

[2] The definition of $\uparrow u$ involves an implicit alphabet $A$ that will always be clear from the context.

When we abstract away from $n$, we said that a language $L \subseteq A^*$ is piecewise-testable (or PT) if it is $n$-PT for some $n$. Other characterisations are: $L$ is PT iff its syntactic monoid is $\mathcal{J}$-trivial (Simon's Theorem), iff it is definable in the $\mathcal{B}\Sigma_1$ fragment of the first-order logic over words [4], iff its canonical DFA is acyclic and locally confluent [16].

Note that if $L$ is $n$-PT, it is also $m$-PT for any $m > n$. We write $h(L)$ for the smallest $n$ —called the "height of $L$"— such that $L$ is $n$-PT, letting $h(L) = \infty$ when $L$ is not PT.

The following properties will be useful:

▸ **Lemma 2.1.** *For all $u, v \in A^*$ and $a \in A$:*
*(1) If $u \lesssim_n v$ then $u \sim_n w$ for all $w \in A^*$ such that $u \sqsubseteq w \sqsubseteq v$.*
*(2) If $u \sim_n v$ then there exists $w \in A^*$ such that $u \lesssim_n w$ and $v \lesssim_n w$.*
*(3) If $uv \sim_n uav$ then $uv \sim_n ua^\ell v$ for all $\ell \in \mathbb{N}$.*
*(4) Every equivalence class of $\sim_n$ is a singleton or is infinite.*

**Proof.** (1) is by combining Eq. (1) and $\downarrow u \subseteq \downarrow w \subseteq \downarrow v$; (2) is Lemma 6 from [29]; (3) is in the proof of [27, Coro. 2.8]; (4) follows from (1), (2) and (3). ◂

**Constructing PT languages.** Recall that PT languages constitute a subvariety of the dot-depth 1 languages, themselves a subvariety of the star-free languages, themselves a subvariety of the regular languages [4]. As such, all classes $\mathsf{PT}_n$ for $n \in \mathbb{N}$, as well as $\mathsf{PT}$, are closed under union, intersection, complementation, inverse morphisms and quotients (left and right residuals). These properties lead to (in)equations like

$$h(L \cup L') \leqslant \max(h(L), h(L'))\,, \qquad h(\neg L) = h(L)\,, \tag{2}$$

$$h(u^{-1}L) \leqslant h(L)\,, \qquad h(Lv^{-1}) \leqslant h(L)\,, \tag{3}$$

$$h(\rho^{-1}(L)) \leqslant h(L) \text{ when } \rho : A^* \to B^* \text{ is a morphism,} \tag{4}$$

that can be used to bound the height of the PT languages we construct. See Appendix B for a proof of Eq. (4).

**Relating PT height and state complexity.** For regular languages, a standard measure of descriptive complexity is *state complexity*, denoted $sc(L)$, and defined as the number of states of the canonical DFA for $L$.

The bounds we just listed let us contrast the height of a PT language with its state complexity. For PT languages, $h(L)$ is smaller than or equal to $sc(L)$ (equality occurs e.g. when $L = \{a^\ell\}$) since $sc(L)$ bounds the depth of the automaton, i.e., the maximum length of a simple path from the initial to some final state, which in turns bounds $h(L)$ [16, 22].

In the other direction, we can prove

▸ **Theorem 2.2.** *Let $A$ be an alphabet of size $k$ with $k > 1$. Suppose $L \subseteq A^*$ is $n$-PT. Then the canonical DFA for $L$ has at most $m$ states,*[3] *where*

$$\log m = k \left( \frac{n + 2k - 3}{k - 1} \right)^{k-1} \log n \log k\,.$$

*Here* log *means* log *to the base* 2. *Thus, for fixed $k$, $sc(L)$ is in $2^{O(n^{k-1} \log n)}$, where $n = h(L)$.*

[3] It is shown in [22] that the depth (not the size) of the canonical DFA is bounded by $\binom{n+k}{n} - 1$.

**Proof.** We build a DFA for $L$ which remembers the equivalence class under $\sim_n$ of the word it has read so far. This is possible because for all $w \in A^*$ and $a \in A$, the class $[wa]_n$ of $wa$ is determined by $[w]_n$ and $a$. The initial state is $[\varepsilon]_n$, and the final states are all the classes $[u]_n$ which are a subset of $L$. In [11] we showed that the number of equivalence classes of $\sim_n$ is bounded by $m$. ◂

The general situation is that $h(L)$ can be much smaller than $sc(L)$ as we see in the following sections. More importantly, $h(L)$ is more robust than $sc(L)$ and, for example, state complexity will usually increase (sometimes exponentially) when constructing a regular language with boolean combinations of simpler languages[4] while PT height will not increase.[5]

**More constructions for PT languages.** Two simple but important constructions that provide PT languages are the closures by subwords and superwords, $\uparrow L$ and $\downarrow L$, defined above. Every upward-closed language is PT since it is the union of finitely many languages of the form $\uparrow u$ (by Higman's Lemma [18]). Every downward-closed language is PT too since its complement is upward-closed. Analysing the height of these PT languages is the topic of Sections 4 and 5.

We are not aware of more piecewise-testability preserving operations on languages in the literature. Let us recall that PT languages are not closed under concatenation (even just $L \mapsto a.L$), Kleene star, shuffle product, conjugacy, and simple operations like renamings (length-preserving morphisms) or the erasing of one letter, see Appendix A for details.

In view of this, it was a good surprise to discover that $I(L)$ is PT when $L$ is. Bounding its height requires a non-trivial ad hoc proof and is the topic of Section 6.

## 3 PT height of words and the small-subword theorem

Our starting point is an analysis of the PT height of single words. It is clear that any singleton language $\{u\}$ is PT since $\{u\} = \uparrow u \smallsetminus \bigcup_{v \in \{u\} \sqcup\!\sqcup A} \uparrow v$, which entails $h(\{u\}) \leqslant |u| + 1$. Here $\{u\} \sqcup\!\sqcup A$ is a *shuffle product*, collecting all the shuffles of $u$ with a letter from $A$. In other words, $\{u\} \sqcup\!\sqcup A = \{v \ : \ u \sqsubseteq v \wedge |v| = |u| + 1\}$. Below we often omit set-theoretical parentheses when denoting singletons, writing e.g. "$h(u)$" or "$u \sqcup\!\sqcup A$".

The PT height of a singleton language can be computed in time $O((|u| + |A|) \cdot |u| \cdot |A|)$, see Appendix C. This can be used to compute the PT height of finite languages: for such languages, the inequality in Eq. (2) becomes

$$h(\{u_1, \ldots, u_m\}) = \max\{h(u_1), \ldots, h(u_m)\} \,. \tag{5}$$

Indeed, $h(\{u_1, \ldots, u_m\}) = n$ implies $[u_i]_n \subseteq \{u_1, \ldots, u_m\}$ for any $i$. Thus $[u_i]_n$ is a singleton in view of Lemma 2.1.4. Hence $[u_i]_n = \{u_i\}$ and $h(u_i) \leqslant n$.

The $|u| + 1$ upper bound for $h(u)$ is reached for $u = a^\ell$ (to see that $h(a^\ell) > \ell$, one notes that $\{a^\ell\}$ is not closed under $\sim_\ell$ since $a^\ell \sim_\ell a^{\ell+1}$). However, words on more than one letter can generally be described within some PT height lower than their length. For example

$$\{aabb\} = (\uparrow aa \cap \uparrow bb) \smallsetminus (\uparrow ba \cup \uparrow aaa \cup \uparrow bbb) \,,$$

[4] Such combinatorial explosions also occur when restricting to piecewise-testable languages [12].

[5] This robustness is not restricted to boolean operations: write $rev(u)$ for the *reversal* of $u$, e.g., $rev(abc) = cba$ and extend to languages. It is clear that $rev(L)$ is $n$-PT when $L$ is —indeed $rev(\uparrow u) = \uparrow rev(u)$, $rev(L \cup L') = rev(L) \cup rev(L')$, and $rev(A^* \smallsetminus L) = A^* \smallsetminus rev(L)$— but $sc(rev(L))$ cannot be bounded by a polynomial of $sc(L)$, even in the case of finite, hence PT, languages [28].

showing $h(aabb) \leqslant 3$. (Note that $h(aabb) > 2$ since $aabbb \sim_2 aabb$.) It turns out that the PT height of words can be much lower than their length as we now show.

**Words with low PT height.** We now introduce a family of words with "low PT height" that will be used repeatedly in later sections. Let $A_k = \{a_1, \ldots, a_k\}$ be a $k$-letter alphabet. We define a word $U_k \in A_k^*$ by induction on $k$ and parameterized by a parameter $\eta \in \mathbb{N}$. We let $U_0 \stackrel{\text{def}}{=} \varepsilon$ and, for $k > 0$, $U_k \stackrel{\text{def}}{=} (U_{k-1}a_k)^\eta U_{k-1}$. For example, for $\eta = 3$ and $k = 2$, one has $U_2 = a_1a_1a_1a_2a_1a_1a_1a_2a_1a_1a_1a_2a_1a_1a_1$.

▸ **Proposition 3.1.** *For $k \geqslant 0$, $|U_k| = (\eta+1)^k - 1$ and $h(U_k) = k\eta + 1$.*

**Proof sketch.** An easy induction on $k$ shows that $|U_k| = (\eta+1)^k - 1$. To show $h(U_k) = k\eta+1$ we use auxiliary languages $P_k, N_k \subseteq A_k^*$ defined inductively by:

$$P_0 = \{\varepsilon\}, \quad N_0 = \varnothing,$$

and, for $k > 0$,

$$P_k = \{a_k^i v a_k^{\eta-i} \mid 0 \leqslant i \leqslant \eta \wedge v \in P_{k-1}\},$$
$$N_k = \{a_k^{\eta+1}\} \cup \{a_k^i w a_k^{\eta-i} \mid 0 \leqslant i \leqslant \eta \wedge w \in N_{k-1}\}.$$

We now claim that for any $k \in \mathbb{N}$ and $u \in A_k^*$:

$$\Big(\bigwedge_{v\in P_k} v \sqsubseteq u\Big) \wedge \Big(\bigwedge_{w\in N_k} w \not\sqsubseteq u\Big) \iff u = U_k. \tag{6}$$

(See Appendix D for a proof.) Thus $h(U_k) \leqslant k\eta + 1$ since the words in $P_k$ have length $k\eta$ and the words in $N_k$ have length at most $k\eta + 1$.

It remains to show that $h(U_k) > k\eta$, i.e., that $\{U_k\}$ is not closed under $\sim_{k\eta}$: for this it is enough to note that $U_k \sim_{k\eta} U_k a_1$ using [29, Lemma 3]. ◂

For later use we also record the following bounds (see Appendix E):

$$h(\downarrow U_k) = \eta(\eta+1)^{k-1} + 1\,. \tag{7}$$

**Rich words and rich factorizations.** Assume a fixed $k$-letter alphabet $A$. We say that a word $u$ is *rich* if all the $k$ letters of $A$ occur in it, and that it is *poor* otherwise. For $\ell \in \mathbb{N}$, we further say that $u$ is $\ell$-rich if it can be written as a concatenation $u = u_1 \cdots u_\ell u'$ where the $\ell$ factors $u_1, \ldots, u_\ell$ are rich.

The *richness* of $u$ is the largest $\ell \in \mathbb{N}$ such that $u$ is $\ell$-rich. Note that having $|u|_a \geqslant \ell$ for all letters $a \in A$ does not imply that $u$ is $\ell$-rich.

▸ **Lemma 3.2** (See Appendix F)**.** *If $u_1$ and $u_2$ are respectively $\ell_1$-rich and $\ell_2$-rich, then $v \sim_n v'$ implies $u_1vu_2 \sim_{\ell_1+n+\ell_2} u_1v'u_2$.*

The *rich factorization* of $u \in A^*$ is the decomposition $u = u_1a_1 \cdots u_ma_mv$ defined by induction in the following way: if $u$ is poor, we let $m = 0$ and $v = u$; otherwise $u$ is rich, we let $u_1a_1$ (with $a_1 \in A$) be the shortest prefix of $u$ that is rich and let $u_2a_2 \cdots u_ma_mv$ be the rich factorization of the remaining suffix $(u_1a_1)^{-1}u$. By construction $m$ is the richness of $u$. E.g., assuming $k = 3$ and $A = \{a, b, c\}$, the following is a rich factorization with $m = 2$:

$$\overbrace{bbaaabbccccaabbbaa}^{u} = \overbrace{bbaaabb}^{u_1} \cdot c \cdot \overbrace{cccaa}^{u_2} \cdot b \cdot \overbrace{bbaa}^{v}$$

Note that, by construction, $u_1, \ldots, u_m$ and $v$ are poor.

▸ **Lemma 3.3** (See Appendix F). *Consider two words $u, u'$ of richness $m$ and with rich factorizations $u = u_1a_1 \cdots u_ma_mv$ and $u' = u'_1a_1 \cdots u'_ma_mv'$. Suppose that $v \sim_n v'$ and that $u_i \sim_{n+1} u'_i$ for all $i = 1, \ldots, m$. Then $u \sim_{n+m} u'$.*

**The small-subword theorem.** Our next result is used to prove lower bounds on the PT height of long words. It will be used repeatedly in the course of this paper.

For $k = 1, 2, \ldots$ define $f_k : \mathbb{N} \to \mathbb{N}$ by induction on $k$ with

$$f_1(n) = n\,, \tag{8}$$

$$f_{k+1}(n) = \max_{0 \leqslant m \leqslant n} mf_k(n+1-m) + m + f_k(n-m)\,. \tag{9}$$

The definition of $f_k$ is only used in the proof of Theorem 3.4. In the rest of the paper, we simplify things by relying on the following upper bound (proved in [13, Prop. 4.4]):

$$f_k(n) \leqslant \Big(\frac{n+2k-1}{k}\Big)^k - 1 < \Big(\frac{n}{k}+2\Big)^k\,. \tag{10}$$

▸ **Theorem 3.4** (Small-subword Theorem). *Let $k = |A|$. For all $u \in A^*$ and $n \in \mathbb{N}$ there exists some $v \in A^*$ with $v \lesssim_n u$ and such that $|v| \leqslant f_k(n)$.*

**Proof.** By induction on $k$, the size of the alphabet.

With the base case, $k = 1$, we consider a unary alphabet $A = \{a\}$ and $u$ is $a^{|u|}$. Now $a^\ell \sim_n u$ iff $\ell = |u| < n$ or $\ell \geqslant n \leqslant |u|$. So taking $v = a^\ell$ for $\ell = \min(n, |u|)$ proves the claim.

When $k > 1$ we consider the rich factorization $u = u_1a_1u_2a_2 \cdots u_ma_mu'$ of $u$. Let $n' = \max(n+1-m, 1)$. Every $u_i$ is a word on the subalphabet $A \setminus \{a_i\}$. Hence by induction hypothesis there exists $v_i \sqsubseteq u_i$ with $|v_i| \leqslant f_{k-1}(n')$ and $v_i \sim_{n'} u_i$, entailing $u_ia_i \sim_{n'} v_ia_i$. Similarly, the induction hypothesis entails the existence of some $v' \sqsubseteq u'$ with $v' \sim_{n'-1} u'$ and $|v'| \leqslant f_{k-1}(n'-1)$. Note that in these inductive steps we use a length bound obtained with $f_{k-1}$ by using the fact that $u_1, \ldots, u_m$ and $u'$, being poor, use at most $k-1$ letters from $A$.

We now consider two cases. If $m \leqslant n-1$, we let $v = v_1a_1 \cdots v_ma_mv'$, so that $v \sqsubseteq u$ and $|v| \leqslant mf_{k-1}(n') + m + f_{k-1}(n'-1)$. We deduce $|v| \leqslant f_k(n)$ using Eq. (9) and since $n' = n+1-m$. That $v \sim_n u$, hence $v \lesssim_n u$, is an application of Lemma 3.3: $v_1a_1 \cdots v_ma_mv'$ is indeed the rich decomposition of $v$ since $n' \geqslant 2$, $v' \sim_{n'-1} u'$, and $v_i \sim_{n'} u_i$ for $i = 1, \ldots, m$.

If $m \geqslant n$, then $u$ is $n$-rich and its subwords include all words of length at most $n$. It is easy to extract some $n$-rich subword $v$ of $u$ that only uses $kn$ letters. Now $v \sim_n u$ since both $u$ and $v$ are $n$-rich, entailing $v \lesssim_n u$. One also checks that $|v| = kn \leqslant f_k(n)$. ◂

Note that the bound $f_k(n)$ in Theorem 3.4 does not depend on $u$.

We can already apply the small-subword theorem to the case of finite languages.

▸ **Proposition 3.5** (Finite languages). *Suppose $L \subseteq A^*$ is finite and nonempty with $|A| = k$. Let $\ell$ be the length of the longest word in $L$. Then $1 + k(\ell^{1/k} - 2) < h(L) \leqslant \ell + 1$.*

**Proof.** Thanks to Eq. (5), it is enough to consider the case where $L = \{u\}$ is a singleton. So assume $h(L) = h(u) = n$ and $|u| = \ell$. The small-subword theorem says that $u \sim_n v$ for some short $v$ but necessarily $v = u$ since $[u]_n$ is a singleton, hence $\ell \leqslant f_k(n)$. Using Eq. (10) one gets $\ell \leqslant f_k(n) < \big(\frac{n+2k-1}{k}\big)^k$. This gives $n > 1 + k(\ell^{1/k} - 2)$. The upper bound $h(L) \leqslant \ell + 1$ was observed earlier. ◂

We already noted that the upper bound is tight. The lower bound is quite good: for $U_k$ seen above, $\ell = (\eta+1)^k - 1$, so that $\ell \leqslant \big(\frac{n+2k-1}{k}\big)^k - 1$ gives $n = h(U_k) \geqslant k\eta - k + 1$ while we know $h(U_k) = k\eta + 1$.

## 4 Upward closures

It is known that $\uparrow L$ is PT for any $L$. Related languages are $\uparrow_< L$ (used in Section 7) and $\min(L) \stackrel{\text{def}}{=} \{u \in L \mid \forall v \in L : v \not\sqsubset u\}$. This section provides bounds on the PT height of languages obtained by such constructions.

We first note that, in the special case of singletons, the PT height of $\uparrow L$ and $I(L)$ always coincide with word length:[6]

▸ **Proposition 4.1.** *For any $u \in A^*$*

$$h(\uparrow u) = |u|\,, \qquad h(I(u)) = h(\uparrow u \cup \downarrow u) = \begin{cases} |u| & \text{if } |A| \geqslant 2, \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

**Proof.** Let $\ell = |u|$. Obviously $h(\uparrow u) \leqslant \ell$ and the point is to prove $h(\uparrow u) > \ell - 1$. For this we assume $\ell > 0$ and write $u = a_1 \cdots a_\ell$. With a letter $a \in A$ we associate a word $\pi_a$ of length $|A|$ that lists all the letters of $A$ exactly once *and ends with* $a$. E.g. $\pi_b = acdb$ works when $A = \{a, b, c, d\}$. Let now $v = \pi_{a_1}\pi_{a_2} \cdots \pi_{a_{\ell-1}}$ and $v' = v \cdot a_\ell$. Then $v \sim_{\ell-1} v'$ since $v$ has all subwords of length $\ell - 1$. However $u \not\sqsubset v$ and $u \sqsubseteq v'$ hence $\uparrow u$ is not closed under $\sim_{\ell-1}$.

Now for $I(u)$: we note that *in the case of singletons* we can write $I(u) = A^* \setminus (\uparrow u \cup \downarrow_< u)$, from which $h(I(u)) \leqslant \ell$ follows since all the finitely many words in $\downarrow_< u$ have length at most $\ell - 1$. To show $h(I(u)) > \ell - 1$ when $|A| \geqslant 2$, we assume $\ell > 1$ and use $v$ and $v'$ again: $v' \notin I(u)$ while $v \in I(u)$ hence $I(u)$ is not closed under $\sim_{\ell-1}$. Finally, when $|A| < 2$ or $\ell = 0$, $I(u) = \varnothing$, while when $\ell = 1$ and $|A| \geqslant 2$, $I(u)$ is neither $\varnothing$ nor $A^*$. ◂

▸ **Corollary 4.2.** *For any $L \subseteq A^*$ and $m \in \mathbb{N}$, if all words in $\min(L)$ have length bounded by $m$, then $\uparrow L$ is $m$-PT while $\uparrow_< L$ and $\min(L)$ are $(m+1)$-PT.*

**Proof.** Recall that $\min(L)$ is finite by Higman's Lemma. Then note that $\uparrow L = \uparrow \min(L)$ and that $\uparrow_< L = (\uparrow L) \setminus \min(L)$. Use $h(u) \leqslant |u| + 1$ from Section 3. ◂

This can be immediately applied to languages given by automata or grammars.

▸ **Theorem 4.3** (Upward closures of regular and context-free languages)**.**
*(1) If $L$ is accepted by a nondeterministic automaton (a NFA) having depth $m$, then $\uparrow L$ is $m$-PT while $\uparrow_< L$ and $\min(L)$ are $(m+1)$-PT.*
*(2) The same holds if $L$ is accepted by a context-free grammar (a CFG) when we let $m = \ell^N$ where $N$ is the number of nonterminal symbols and $\ell$ is the maximum length for the right-hand side of production rules.*

**Proof sketch.** (1) A word accepted by the NFA is minimal wrt $\sqsubseteq$ only if it is accepted along an acyclic path. (2) A word generated by the CFG is minimal wrt $\sqsubseteq$ only if any nonterminal appears at most once along any branch of its derivation tree. ◂

The bounds in Theorem 4.3 can be reached, e.g., for $L$ a singleton of the form $\{a^m\}$.

For our applications, we are interested in expressing $h(\uparrow L)$ in terms of $h(L)$, assuming that $L$ is PT.

▸ **Theorem 4.4** (Upward closures of PT languages)**.** *Suppose $L \subseteq A^*$ is $n$-PT and $|A| = k$. Let $m = f_k(n)$. Then $\uparrow L$ is $m$-PT, while $\uparrow_< L$ and $\min(L)$ are $(m+1)$-PT.*

[6] This phenomenon does not extend to the other operations nor to finite sets.

**Proof.** By the small-subword theorem, and since $L$ is closed under $\sim_n$, the minimal elements of $L$ have length bounded by $m$. Then Corollary 4.2 applies. ◂

▸ **Remark 4.5.** *The upper bound in Theorem 4.4 is quite good: for any $k, \eta \geqslant 1$, the language $L = \{U_k\}$ has $h(U_k) = n = k\eta + 1$ so that Theorem 4.4 with Eq. (10) give $h(\uparrow U_k) \leqslant f_k(n) < (\eta+2)^k$. On the other hand we know that $h(\uparrow U_k) = (\eta+1)^k - 1$ by Proposition 4.1.*

## 5 Downward closures

We now move to downward closures. It is known that, for any $L \subseteq A^*$, $\downarrow L$ and $\downarrow_< L$ are PT since they are the complement of upward-closed languages. Our strategy for bounding $h(\downarrow L)$ is to approximate $L$ by finitely many D-products.

A *D-product* is a regular expression $P$ of the form $E_1 \cdot E_2 \cdots E_\ell$ where every $E_i$ is either of the form $B^*$ for a subalphabet $B \subseteq A$ ($B^*$ is called a *star factor* of $P$), or a single letter $a \in A$ (a *letter factor*). We say that $\ell$ is the length of $P$.

▸ **Proposition 5.1.** *If $P$ is a D-product of length $\ell$, $h(\downarrow P) \leqslant \ell + 1$ and $h(\downarrow_< P) \leqslant \ell + 1$.*

**Proof.** Let $P'$ be the regular expression obtained from $P$ by replacing any letter factor $a$ by $(a + \varepsilon)$ so that $P' = \downarrow P$. Now any residual $w^{-1}P'$ of $P'$ is either the empty language $\varnothing$, or corresponds to a suffix $P''$ of $P'$. This is shown by induction on the length of suffixes, using

$$b^{-1}\big[(a+\varepsilon)P''\big] = \begin{cases} P'' & \text{if } b = a, \\ b^{-1}P'' & \text{otherwise,} \end{cases} \qquad b^{-1}\big[B^*P''\big] = \begin{cases} B^*P'' & \text{if } b \in B, \\ b^{-1}P'' & \text{otherwise,} \end{cases}$$

and $a^{-1}\varepsilon = \varnothing$ for the last suffix, i.e., the empty product. (Note that the correctness of the first equality when $b = a$, and of the second equality when $b \in B$, rely on $b^{-1}P'' \subseteq P''$: this holds because $P''$ is downward-closed.) Thus $P'$ has at most $\ell + 1$ distinct non-empty residuals, i.e., the canonical DFA for $P'$ has at most $\ell+1$ productive states, hence has depth at most $\ell + 1$. We now apply Theorems 1 and 2 from [16] and conclude that $h(\downarrow P) \leqslant \ell + 1$.

For $\downarrow_< P$ very little need to be changed. If $P$ contains at least one star factor then $\downarrow_< P$ and $\downarrow P$ coincide. If $P$ only contains letter factors then $P$ denotes a singleton $\{u\}$ with $|u| = \ell$. Then $\downarrow_< P$ is a finite set of words of length at most $\ell - 1$, entailing $h(\downarrow_< P) \leqslant \ell$. ◂

The bounds in Proposition 5.1 can be reached, e.g., for $P = a \cdots a$.

▸ **Corollary 5.2.** *If $L \subseteq \bigcup_i P_i \subseteq \downarrow L$ for a family $(P_i)_i$ of D-products of length at most $\ell$, then $h(\downarrow L) \leqslant \ell + 1$ and $h(\downarrow_< L) \leqslant \ell + 1$.*

**Proof.** Obviously $\downarrow L = \bigcup_i \downarrow P_i$ and the union is finite since there are only finitely many D-products of bounded length. ◂

This can be immediately applied to languages given by automata or grammars.

▸ **Theorem 5.3** (Downward-closures of regular and context-free languages)**.**
*(1) If $L$ is accepted by a nondeterministic automaton (a NFA) having depth $m$, then $\downarrow L$ and $\downarrow_< L$ are $\ell$-PT for $\ell = 2m + 2$.*
*(2) The same holds if $L$ is accepted by a CFG in quadratic normal form (a QNF, see [1]) with $N$ nonterminals and $\ell = 4 \cdot 3^{N-1} + 2$.*

**Proof sketch.** (1) For a word $u \in L$ we consider the cycles in an accepting path on $u$. This leads to a factoring $u = u_0 a_1 u_1 a_2 \cdots a_p u_p$ of $u$ such that the accepting path is some $q_0 \xrightarrow{u_0} q_0 \xrightarrow{a_1} q_1 \xrightarrow{u_1} q_1 \xrightarrow{a_2} q_2 \cdots q_{p-1} \xrightarrow{a_p} q_p \xrightarrow{u_p} q_p$ with $q_0, q_1, \ldots, q_p$ all different. Then $p \leqslant m$. Let

now $B_i \subseteq A$ be the set of letters occurring in $u_i$ and define $P_u \stackrel{\text{def}}{=} B_0^* a_1 B_1^* a_2 \dots B_{p-1}^* a_p B_p^*$. Then $u \in P_u$ and $P_u \subseteq {\downarrow}L$. Finally, $L \subseteq \bigcup_{u \in L} P_u \subseteq {\downarrow}L$ and each $P_u$ has length $\leqslant 2m+1$.
(2) Bachmeier et al. showed that there is an NFA for ${\downarrow}L$ having $2 \cdot 3^{N-1}$ states [1]. ◂

For our applications, we are interested in bounding $h({\downarrow}L)$ in terms of $h(L)$ when $L$ is PT.

▸ **Theorem 5.4** (Downward closures of PT languages). *Suppose $L \subseteq A^*$ is $n$-PT and $|A| = k$. Let $m = f_k(n)$. Then ${\downarrow}L$ and ${\downarrow_<}L$ are $(k+1)(m+1)$-PT.*

The rest of this section is devoted to the proof of Theorem 5.4. Our strategy is to approximate $L$ by D-products, this time relying on the fact that $L$ is closed under $\sim_n$.

Recall Lemma 2.1.3 stating that, given two words $u, v$ and a letter $a \in A$, $uv \sim_n uav$ entails $uv \sim_n ua^\ell v$ for all $\ell \in \mathbb{N}$. We express this as "$uav \in [uv]_n$ implies $ua^*v \subseteq [uv]_n$" and call it a *pumping property* of PT classes. We now establish more general pumping properties.

▸ **Lemma 5.5.** *If $uB^*C^*B^*v \subseteq [uv]_n$, where $B, C \subseteq A$ are subalphabets, then $u(B \cup C)^*v \subseteq [uv]_n$.*

**Proof idea.** We prove that for any $m \in \mathbb{N}$, for any $w \in B^*(C^*B^*)^m$, for any $s \in A^{\leqslant n}$, $s \sqsubseteq uwv$ implies $s \sqsubseteq uv$. The proof is by induction on $m$, knowing that the claim holds by assumption for $m \leqslant 1$. See Appendix G for details. ◂

Going on we can show that $uab_1b_2 \cdots b_m av \sim_n uv$ entails $uwv \sim_n uv$ for all $w \in (a + b_1)^*(a + b_2)^* \cdots (a + b_m)^*$, hence the two surrounding $a$'s can join any surrounded letter.

▸ **Lemma 5.6** (See Appendix G). *Suppose $L_1 B_1^* B_2^* \cdots B_\ell^* L_2 \subseteq [u]_n$ for some languages $L_1, L_2 \subseteq A^*$ and subalphabets $B_1, B_2, \dots, B_\ell \subseteq A$ with $\ell \geqslant 3$. If $a \in B_1 \cap B_\ell$ then, letting $B'_i = B_i \cup \{a\}$, $L_1 B'^*_1 B'^*_2 \cdots B'^*_\ell L_2 \subseteq [u]_n$.*

There remains to bound the products that cannot be simplified by the above Lemmas.

▸ **Lemma 5.7.** *Suppose $A$ is a finite set and $E_1, E_2, \dots, E_\ell$ are $\ell > 1$ subsets of $A$ such that the following hold:*
*— for all $1 \leqslant i < \ell$, $E_i \not\subseteq E_{i+1}$ and $E_{i+1} \not\subseteq E_i$ ;*
*— for all $a \in A$ and $1 \leqslant i < j \leqslant \ell$, if $a \in E_i \cap E_j$, then $a \in E_k$ for all $i \leqslant k \leqslant j$.*
*Then $\ell \leqslant |A|$.*

**Proof.** Note that by the first condition, each $E_i$ is nonempty. Define $E_0 = E_{\ell+1} = \varnothing$. For $0 \leqslant i \leqslant \ell$, define $F_i = E_i \mathbin{\triangle} E_{i+1}$, where $\triangle$ denotes symmetric difference. Now $F_0$ and $F_\ell$ have size at least 1, and by the first condition, every other $F_i$ has size at least 2. Thus $\sum_i |F_i| \geqslant 2\ell$. By the second condition, any $a \in A$ occurs in at most two $F_i$'s, thus $\sum_i |F_i| \leqslant 2|A|$. So we conclude $2\ell \leqslant 2|A|$. ◂

▸ **Lemma 5.8.** *Let $L \subseteq A^*$ be $n$-PT. Let $k = |A|$ and $m = f_k(n)$. For every $u \in L$ there is a D-product $P_u$ of length $\ell \leqslant mk + m + k$ such that $u \in P_u \subseteq L$.*

**Proof idea.** A formal proof is given in Appendix G and we just outline it here: Assume $u \in L$. By the small-subword theorem there exists a subword $v = a_1 \cdots a_\ell$ of $u$ with $v \sim_n u$ and $\ell = |v| \leqslant m$. So $u$ is $v$ plus some added letters. By Lemma 2.1.3, all these added (occurrences of) letters can be pumped, yielding a D-product with $u \in P \subseteq [u]_n$. Applying Lemma 5.6 and further simplifications yields a shorter D-product with $P \subseteq P_u \subseteq [u]_n$. Since $P_u$ cannot be further simplified, Lemma 5.7 can be used to bound its size. ◂

We may now conclude:

**Proof of Theorem 5.4.** With Lemma 5.8 we obtain $L = \bigcup_{u \in L} P_u$ where each $P_u$ has length bounded by $km + k + m$. We can then apply Proposition 5.1. ◂

▸ **Remark 5.9.** *The upper bound in Theorem 5.4 is quite good: for any $k, \eta \geqslant 1$, the language $L = \{U_k\}$ from Proposition 3.1 has $h(U_k) = n = k\eta + 1$ so that Theorem 5.4 gives $h(\downarrow U_k) < (k+1)(\eta+2)^k$. On the other hand we know that $h(\downarrow U_k) = \eta(\eta+1)^{k-1} + 1$ by Eq. (7).*

## 6 Piecewise-testability and PT height for $I(L)$

Recall that $I(L)$ is the set of words which are incomparable (under $\sqsubseteq$) with *some word* in $L$. In this section we prove the following result.

▸ **Theorem 6.1.** *Suppose $L \subseteq A^*$ is $n$-PT and $|A| = k$. Let $m = f_k(n)$. Then $I(L)$ is $(m+1)$-PT.*

It is not too difficult to show the regularity of $I(L)$ when $L$ is regular, and this can be done using standard automata-theoretical techniques. Indeed, it can be shown that the incomparability relation $\perp$ is a rational relation [14].

Showing that $I$ also preserves piecewise-testability requires more work. For such questions, $I$ does not behave as simply as the other pre-images we considered before. In particular, we observe that $I(L)$ is not necessarily PT when $L$ is regular. For example, taking $A = \{a, b, c\}$ and letting

$$L = (abc)^*(\varepsilon + a + ab) = \{\varepsilon, a, ab, abc, abca, abcab, \ldots\}$$

gives a language that is totally ordered by $\sqsubseteq$ and contains one word of each length, so that $I(L) = A^* \smallsetminus L$, which is not PT since $L$ is not.

Similarly, $I(L)$ is not necessarily regular when $L$ is not. For example, taking $A = \{a, b\}$ and

$$L = \{a^\ell b^\ell(\varepsilon + b) \mid \ell \in \mathbb{N}\} = \{\varepsilon, b, ab, abb, aabb, a^2b^3, a^3b^3, \ldots\}\,.$$

Again $L$ is totally ordered by $\sqsubseteq$ and contains one word of each length. Hence $I(L) = A^* \smallsetminus L$, which is not regular.

The above examples illustrate our strategy for proving Theorem 6.1: if a language $L$ is totally ordered by $\sqsubseteq$ then $I(L) \cap L = \varnothing$, or equivalently $I(L) \subseteq A^* \smallsetminus L$. Similarly, if $L$ contains at least two words having same length $\ell$ then $I(L)$ contains all words of length $\ell$.

We now proceed with a more formal proof. For technical convenience we introduce a dual construct:

$$C(L) \stackrel{\text{def}}{=} \{u \in A^* \mid L \subseteq \uparrow u \cup \downarrow u\}\,.$$

Note that $C(L)$ coincides with $A^* \smallsetminus I(L)$ and that $C(L \cup L') = C(L) \cap C(L')$. We find it easier to analyse $C(L)$ instead of $I(L)$, but these two languages have the same PT height.

As we just hinted at, it is useful to think of the "layers" $L \cap A^{=\ell} = \{w \in L : |w| = \ell\}$ of $L$, and check whether they contain 0, 1 or more words (we say that the layer is *empty*, *singular*, or *populous*). Observe that if $L \cap A^{=\ell}$ is populous then $C(L) \cap A^{=\ell}$ is empty.

For the rest of this section, we consider a fixed $n \geqslant 1$ and let $m = f_k(n)$. We start with a technical lemma: write $u \lesssim^1_n v$ when $u \lesssim_n v$ and $|v| = |u| + 1$, i.e., $v$ is $u$ with one letter added in a way that is compatible with $\sim_n$. Note that $\lesssim_n$ is the transitive closure of $\lesssim^1_n$.

▸ **Lemma 6.2.** *Let $u, v, w \in A^*$ such that $u \lesssim_n^1 w$ and $v \lesssim_n^1 w$ with $u \neq v$. Then there exists $w' \in A^*$ with $|w'| = |w|$, $w' \neq w$, and $w \sim_n w'$.*

**Proof idea.** Since $|u| = |v| = |w| - 1$, $w$ must be some $w_0a_1w_1a_2w_2$ with $a_1, a_2 \in A$ such that $u = w_0a_1w_1w_2$ and $v = w_0w_1a_2w_2$. We claim that $w' \stackrel{\text{def}}{=} w_0w_1a_2a_2w_2$ witnesses the lemma. Since $u \neq v$, we have $a_1w_1 \neq w_1a_2$, and thus $w \neq w'$. There remains to show that $w \sim_n w'$: this is done by a standard case analysis, see Appendix H. ◂

In the rest of this section, we consider some $\sim_n$-class $T \subseteq A^*$. The populous layers of $T$ propagate upwards:

▸ **Lemma 6.3.** *If $T \cap A^{=p}$ is populous, then $T \cap A^{=p+1}$ is populous too.*

**Proof.** Suppose that $T$ contains two distinct words $u$ and $v$ of length $p$. Then there is some $w$ with $u \lesssim_n w \gtrsim_n v$ (Lemma 2.1.2) hence some $u', v'$ with $u \lesssim_n^1 u'$ and $v \lesssim_n^1 v'$ (Lemma 2.1.1). If $u' \neq v'$ we are done since $|u'| = |v'| = p + 1$. If $u' = v'$ then $u \lesssim_n^1 u' \gtrsim_n^1 v$ and Lemma 6.2 shows that $T$ contains at least two words of length $p + 1$. ◂

Populous layers also propagate downwards in the following sense:

▸ **Lemma 6.4.** *Let $p$ be the length of the shortest word in $T$ and suppose that $T \cap A^{=q}$ is populous, for some $q > p$. Then $T \cap A^{=p+1}$ is populous.*

**Proof.** Let $q$ be the smallest layer such that $T \cap A^{=q}$ is populous. If $q = p + 1$ we are done, and similarly if $q = p$ (Lemma 6.3). So assume $q \geqslant p + 2$. For all $\ell$ with $p \leqslant \ell < q$, the layers $T \cap A^{=\ell}$ are nonempty (by Lemma 2.1) hence singular. Further, Lemma 2.1 tells us the form of the words in these layers: $T \cap A^{<q} = \{uv, uav, \ldots, ua^{q-p-1}v\}$ for some $u, v \in A^*$ and $a \in A$.

We now turn to $T \cap A^{=q}$. This populous layer contains some word $w \neq ua^{q-p}v$. By Theorem 6.2.9 of [27], all minimal (with respect to $\sqsubseteq$) words of $T$ have the same length, hence $w$ is not minimal in $T$, and is obtained by inserting a single letter in $ua^{q-p-1}v$. Define a word $s$ as follows, depending on $w$:

- If $w = u'a^{q-p-1}v$ with $u' \sqsupseteq u$ and $|u'| = |u| + 1$, then $s = u'v$.
- If $w = u'a^\ell ba^{q-p-1-\ell}v$ with $b \neq a$, then $s = ubv$.
- If $w = ua^{q-p-1}v'$ with $v' \sqsupseteq v$ and $|v'| = |v| + 1$, then $s = uv'$.

The idea is that $s$ is obtained by adding a letter to $uv$ "exactly like" $w$ is obtained from $ua^{q-p-1v}$. Since $w \neq ua^{q-p}v$, it is easy to see that $s \neq uav$. Since $uv \sqsubseteq s \sqsubseteq w$ and $uv \sim_n w$, we have $uv \sim_n s \sim_n w$. Thus $T$ has at least two words of length $p + 1$, namely $uav$ and $s$. ◂

We now handle a special case:

▸ **Lemma 6.5.** *If $T$ is not linearly ordered by $\sqsubseteq$, then $C(T)$ is finite, and is in fact a subset of $A^{\leqslant m}$.*

**Proof.** Assume $T$ is not linearly ordered by $\sqsubseteq$ and pick $u, v \in T$ with $u \not\sqsubseteq v$ and $|u| \leqslant |v|$. Let $q \stackrel{\text{def}}{=} |v|$. By Lemma 2.1, there exists $w \in T$ such that $u \lesssim_n w$ and $v \lesssim_n w$. By Lemma 2.1.2, there exists a $v' \in A^{=q}$ with $u \lesssim_n v' \lesssim_n w$. Furthermore, $v' \neq v$ since $u \not\sqsubseteq v$ and $u \sqsubseteq v'$. Thus $T \cap A^{=q}$ is populous. Since by the small-subword theorem the shortest word in $T$ has length at most $m$, we conclude by Lemmas 6.4 and 6.3 that $T \cap A^{=p}$ is populous for every $p > m$. Thus $C(T) \subseteq A^{\leqslant m}$. ◂

We now consider the general case:

▸ **Lemma 6.6.** *$I(T)$ is $(m+1)$-PT.*

**Proof.** Recall that $T$ is a singleton or is infinite (Lemma 2.1.4). We consider three cases.

- Suppose $T$ is a singleton, $T = \{u\}$. By the small-subword theorem, $|u| \leqslant m$. Then $\uparrow u$ is $m$-PT, and $\downarrow u$ is $(m+1)$-PT. Thus $I(u) = A^* \smallsetminus (\uparrow u \cup \downarrow u)$ is $(m+1)$-PT.
- Suppose $T$ is not a total order under $\sqsubseteq$. Then by Lemma 6.5, $C(T) \subseteq A^{\leqslant m}$, so $C(T)$ is $(m+1)$-PT, and so is $I(T)$.
- Suppose $T$ is infinite and a total order under $\sqsubseteq$. Let $p$ be the length of the shortest word in $T$. By the small-subword theorem, $p \leqslant m$. Since $T$ is infinite, and by Lemma 2.1.2, $T \cap A^{=q}$ is nonempty for every $q \geqslant p$. Since $T$ is a total order under $\sqsubseteq$, none of these $T \cap A^{=q}$ is populous, hence they are all singular. Therefore $C(T) \cap A^{\geqslant p} = T$. It remains to describe $C(T) \cap A^{\leqslant p}$, and this is $\downarrow u_0$, where $u_0$ is the unique word of length $p$ in $T$. Thus $C(T) = T \cup \downarrow u_0$ is $(p+1)$-PT, hence also $(m+1)$-PT, and $I(T)$ too is $(m+1)$-PT. ◂

We may now conclude:

**Proof of Theorem 6.1.** Being $n$-PT, $L$ is a finite union $T_1 \cup \cdots \cup T_\ell$ of equivalence classes of $\sim_n$, so that $I(L) = I(T_1) \cup \cdots \cup I(T_\ell)$. Now each $I(T_i)$ is $(m+1)$-PT by Lemma 6.6 so that $I(L)$ is too. ◂

▸ **Remark 6.7.** *The upper bound in Theorem 6.1 is quite good: for any $k, \eta \geqslant 1$, the language $L = \{U_k\}$ from Proposition 3.1 has $h(U_k) = n = k\eta+1$ so that Theorem 6.1 gives $h(I(U_k)) \leqslant (\eta+2)^k$. On the other hand we know by Eq. (11) that $h(I(U_k)) = |U_k| = (\eta+1)^k - 1$ when $k > 1$.*

# 7 Deciding the two-variable logic of subwords

We assume familiarity with basic notions of first-order logic as exposed in, e.g., [7]: bound and free occurrences of variables, quantifier depth of formulae, and fragments $\mathsf{FO}^n$ where at most $n$ different variables (free or bound) are used.

The signature of the $\mathsf{FO}(A^*, \sqsubseteq)$ logic consists of only one predicate symbol "$\sqsubseteq$", denoting the subword relation. Terms are variables taken from a countable set $X = \{x, y, z, \ldots\}$ and all words $u \in A^*$ as constant symbols (denoting themselves). For example, with $A = \{a, b, c, \ldots\}$, $\exists x\big(ab \sqsubseteq x \wedge bc \sqsubseteq x \wedge \neg(abc \sqsubseteq x)\big)$ is a true sentence as witnessed by $x \mapsto bcab$.

**On motivations.** Logics of sequences usually do not include the subsequence predicate and rather consider the prefix ordering, and/or functions for taking contiguous subsequences or computing the length of sequences, see, e.g., [5, 10]. However, in automated deduction, and specifically in ordered constraints solving, the decidability of logics of simplification orderings on strings and trees —$\mathsf{FO}(A^*, \sqsubseteq)$ being a special case— is a key issue [3, 19]. These works often limit their scope to $\Sigma_1$ or similar fragments since decidability is elusive in this area.

## 7.1 Decidability for $\mathsf{FO}^2(A^*, \sqsubseteq)$

In [14] we showed that validity and satisfiability are decidable for the $\mathsf{FO}^2$ fragment of the logic of subwords (note that the $\mathsf{FO}^3 \cap \Sigma_2$ fragment is undecidable [19, 14]). Since below we use our results on the heights of PT languages to prove a new complexity upper bound on the underlying algorithm, we first need to recall the main lines of the decidability proof (see [14] for full details).

When describing the decision procedure for the $\mathsf{FO}^2$ fragment, it is convenient to enrich the basic logic by allowing all regular expressions as monadic predicates (with the expected

semantics) and we shall temporarily adopt this extension. For example, we can state that the downward closure of $(ab)^*$ is exactly $(a+b)^*$ with $\forall x\big[x \in (a+b)^* \iff \exists y(y \in (ab)^* \wedge x \sqsubseteq y)\big]$.

In the following we consider $\mathsf{FO}^2$ formulae using only $x$ and $y$ as variables. We consider a variant of the logic where we use the binary relations $\sqsubset$, $\sqsupset$, $=$ and $\perp$ instead of $\sqsubseteq$. This will be convenient later. The two variants are equivalent, even when restricting to $\mathsf{FO}^m$ fragments, since the new set of predicates can be defined in terms of $\sqsubseteq$ and vice versa. To simplify notation, we sometimes use negated predicate symbols as in $x \not\sqsubseteq y$ or $x \notin (ab)^*$ with obvious meaning.

▸ **Lemma 7.1.** *Let $\phi(x)$ be an $\mathsf{FO}^2(A^*, \sqsubseteq)$ formula with at most one free variable. Then there exists a regular language $L_\phi \subseteq A^*$ such that $\phi(x)$ is equivalent to $x \in L_\phi$. Furthermore, $L_\phi$ can be built effectively from $\phi$ and $A$.*

**Proof.** By structural induction on $\phi(x)$. If $\phi(x)$ is an atomic formula of the form $x \in L$, the result is immediate. If $\phi(x)$ is an atomic formula that uses a binary predicate, the fact that it has only one free variable means that $\phi(x)$ is a trivial $x = x$, $x \sqsubset x$, $x \sqsupset x$ or $x \perp x$, so that $L_\phi$ is $A^*$ or $\varnothing$.

For formulae of the form $\neg\phi'(x)$ or $\phi_1(x) \vee \phi_2(x)$, we use the induction hypothesis and the fact that regular languages are (effectively) closed under boolean operations.

The remaining case is when $\phi(x)$ has the form $\exists y \phi'(x, y)$. Using the induction hypothesis, we replace any subformulae of $\phi'$ having the form $\exists x\, \psi(x, y)$ or $\exists y\, \psi(x, y)$ with equivalent formulae of the form $y \in L_\psi$ or $x \in L_\psi$ respectively, for appropriate languages $L_\psi$. Now $\phi'$ is quantifier-free. We further rewrite it by pushing all negations inside with the following meaning-preserving rules:

$$\neg(\psi_1 \vee \psi_2) \to \neg\psi_1 \wedge \neg\psi_2\,, \qquad \neg(\psi_1 \wedge \psi_2) \to \neg\psi_1 \vee \neg\psi_2\,, \qquad \neg\neg\psi \to \psi\,,$$

and then eliminating negations completely with:

$$\neg(z \in L) \to z \in (A^* \smallsetminus L)\,, \qquad \neg(z_1\ R_1\ z_2) \to z_1\ R_2\ z_2 \vee z_1\ R_3\ z_2 \vee z_1\ R_4\ z_2\,,$$

where $R_1, R_2, R_3, R_4$ is any permutation of $\mathcal{R} \stackrel{\text{def}}{=} \{=, \sqsubset, \sqsupset, \perp\}$. This last rewrite rule is correct since the four relations form a partition of $A^* \times A^*$: for all $u, v \in A^*$, exactly one of $u = v$, $u \sqsubset v$, $u \sqsupset v$, and $u \perp v$ holds.

Thus, we may now assume that $\phi'$ is a positive boolean combination of atomic formulae. We write $\phi'$ in disjunctive normal form, that is, as a disjunction of conjunctions of atomic formulae. Observing that $\exists y(\phi_1 \vee \phi_2)$ is equivalent to $\exists y\, \phi_1 \vee \exists y\, \phi_2$, we assume w.l.o.g. that $\phi'$ is just a conjunction of atomic formulae. Any atomic formula of the form $x \in L$, for some $L$, can be moved outside the existential quantification, since $\exists y(x \in L \wedge \psi)$ is equivalent to $x \in L \wedge \exists y\, \psi$. All atomic formulae of the form $y \in L$ can be combined into a single one, since regular languages are closed under intersection.

Finally we may assume that $\phi'(x, y)$ is a conjunction of a single atomic formula of the form $y \in L$ (if no such formula appears, we can write $y \in A^*$), and some combination of atomic formulae among $x \sqsubset y$, $x \sqsupset y$, $x = y$, and $x \perp y$. If at least two of these appear, then their conjunction is unsatisfiable, and so $\phi(x)$ is equivalent to $x \in \varnothing$. If none of them appear, $\exists y(y \in L)$ is equivalent to $x \in A^*$ (or to $x \in \varnothing$ if $L$ is empty). If exactly one of them appears, say $x\ R\ y$, then $\exists y\ (y \in L \wedge xRy))$ is equivalent to $x \in L_\phi$ for $L_\phi = R^{-1}(L)$. Now the pre-image $R^{-1}(L)$ is regular and effectively computable from $L$ since all the relations in

$\mathcal{R}$ are rational relations.[7] ◂

▸ **Corollary 7.2.** [14]. *The truth problem for* $\mathsf{FO}^2(A^*, \sqsubseteq)$ *is decidable.*

**Proof.** Lemma 7.1 provides a recursive procedure for computing $L_\phi$, the set of words that make $\phi(x)$ true. When $\phi$ is a closed formula, it is true iff $L_\phi$ is $A^*$. ◂

**Complexity for $\mathsf{FO}^2(A^*, \sqsubseteq)$.** The algorithm underlying the proof of Lemma 7.1 can be implemented using finite-state automata to handle the regular languages $L_\phi$ that are constructed for each subformula. However, steps like complementation or even computing the pre-images $\uparrow_< L$ and $\downarrow_< L$ are costly and may incur an exponential blowup, and this cannot be avoided by using nondeterministic or alternating automata instead of standard deterministic automata [12]. The consequence is that the only clear upper bound for the algorithm is a tower of exponentials whose height is given by the quantifier depth of the formula at hand, hence a nonelementary complexity. Regarding lower bounds, only PSPACE-hardness has been established [14] and we conjecture that $\mathsf{FO}^2(A^*, \sqsubseteq)$ can be decided with elementary complexity.

## 7.2 Complexity of the $\mathsf{FO}^2(A^*, \sqsubseteq)$ logic without regular predicates

It turns out that when regular predicates are not allowed (i.e., when we use the basic logic), the quantifier-elimination procedure will only produce membership constraints $x \in L$ or $y \in L'$ involving PT languages. Furthermore, it is possible to bound the PT height of the defined languages and deduce an elementary complexity upper bound.

▸ **Theorem 7.3** ($\mathsf{FO}^2(A^*, \sqsubseteq)$ has elementary complexity)**.** *If $\phi(x)$ is an $\mathsf{FO}^2$ formula without regular predicates, then $L_\phi$ is a piecewise-testable language with $h(L_\phi)$ in $2^{2^{O(|\phi|)}}$. Furthermore, computing a canonical DFA for $L_\phi$ (hence deciding the truth of $\phi$) can be done in* 3−EXPTIME.

**Proof.** We mimic the proof of Lemma 7.1. In this process we can allow atomic formulae "$x \in L$" when $L$ is PT, since this can be expressed as a boolean combination of atomic formulae of the form $w \sqsubseteq x$. The key extra ingredient is that the pre-images $R^{-1}(L)$ preserve piecewise-testability and that $h(R^{-1}(L))$ is in $O(h(L)^{|A|})$: we invoke Theorem 4.4 for $R = \sqsupset$, Theorem 5.4 for $R = \sqsubset$, and Theorem 6.1 for $R = \perp$.

Finally, when the PT height of $L_\phi$ (and of all intermediary $L_\psi$) have been bounded in $2^{2^{O(|\phi|)}}$, we obtain a bound on the size of the DFAs and the time and space needed to compute them using Theorem 2.2. ◂

# 8 Concluding remarks

We developed several new techniques for proving upper and lower bounds on the PT height of languages constructed by closing w.r.t. the subword ordering or its inverse. We also considered related constructions like taking minimal elements, or taking the image by the incomparability relation. In general, the PT height of upward closures is bounded with the length of minimal words. For downward closures, we developed techniques for expressing them with D-products and bounding their lengths. We illustrated these techniques with

[7] This is well known and easy to see for $\sqsubset$ and $\sqsupset$. It is proved in [14] for $\perp$.

regular and context-free languages but more classes can be considered [31]. More importantly, the closures of PT languages have PT height bounded polynomially in terms of the PT height of the argument. Our main tool here is the small-subword theorem that provides tight lower bounds on the PT height of finite languages, with ad hoc developments for $I(L)$.

These results are used to bound the complexity of the two-variable logic of subwords but we believe that the PT hierarchy can be used more generally as an effective measure of descriptive complexity. (The same can be said of the hierarchies of locally-testable languages, or of dot-depth-one languages).

This research program raises many interesting questions, such as connecting PT height and other measures, narrowing the gaps remaining in our Theorems 4.4, 5.4, and 6.1, and enriching the known collection of PT-preserving operations.

These questions will probably require new insights in PT languages. For example, the experiments we conducted suggest that $h(u \sqcup\!\sqcup v)$ can be bounded by $h(u) + h(v)$ when $u, v$ are words, however we do not know how to prove this. Similarly, we can prove that $L \sqcup\!\sqcup u$ is PT when $L$ is PT and $u$ is a word, but we only have a very tedious proof. We mention these questions since $L \sqcup\!\sqcup A$ is the pre-image of $L$ by $\sqsupseteq^1$ and it seems that the decidability of $\mathsf{FO}^2(A^*, \sqsubseteq)$ can be extended to $\mathsf{FO}^2(A^*, \sqsubseteq, \sqsubseteq^1)$ [20].

## References


1 G. Bachmeier, M. Luttenberger, and M. Schlund. Finite automata for the sub- and superword closure of CFLs: Descriptional and computational complexity. In *Proc. LATA 2015*, volume 8977 of *Lecture Notes in Computer Science*, pages 473–485. Springer, 2015. doi:10.1007/978-3-319-15579-1_37.

2 M. Bojańczyk, L. Segoufin, and H. Straubing. Piecewise testable tree languages. *Logical Methods in Comp. Science*, 8(3), 2012. doi:10.2168/LMCS-8(3:26)2012.

3 H. Comon. Solving symbolic ordering constraints. *Int. J. Foundations of Computer Science*, 1(4):387–412, 1990. doi:10.1142/S0129054190000278.

4 V. Diekert, P. Gastin, and M. Kufleitner. A survey on small fragments of first-order logic over finite words. *Int. J. Foundations of Computer Science*, 19(3):513–548, 2008. doi:10.1142/S0129054108005802.

5 V. Ganesh, M. Minnes, A. Solar-Lezama, and M. C. Rinard. Word equations with length constraints: What's decidable? In *Proc. HVC 2012*, volume 7857 of *Lecture Notes in Computer Science*, pages 209–226. Springer, 2013. doi:10.1007/978-3-642-39611-3_21.

6 L. H. Haines. On free monoids partially ordered by embedding. *Journal of Combinatorial Theory*, 6(1):94–98, 1969. doi:10.1016/S0021-9800(69)80111-0.

7 J. Harrison. *Handbook of Practical Logic and Automated Reasoning*. Cambridge University Press, 2009. doi:10.1017/CBO9780511576430.

8 G. Higman. Ordering by divisibility in abstract algebras. *Proc. London Math. Soc. (3)*, 2(7):326–336, 1952. doi:10.1112/plms/s3-2.1.326.

9 P. Hofman and W. Martens. Separability by short subsequences and subwords. In *Proc. ICDT 2015*, volume 31 of *Leibniz International Proceedings in Informatics*, pages 230–246, 2015. doi:10.4230/LIPIcs.ICDT.2015.230.

10 P. Hooimeijer and W. Weimer. StrSolve: solving string constraints lazily. *Autom. Softw. Eng.*, 19(4):531–559, 2012. doi:10.1007/s10515-012-0111-x.

11 P. Karandikar, M. Kufleitner, and Ph. Schnoebelen. On the index of Simon's congruence for piecewise testability. *Information Processing Letters*, 115(4):515–519, 2015. doi:10.1016/j.ipl.2014.11.008.

**12** P. Karandikar, M. Niewerth, and Ph. Schnoebelen. On the state complexity of closures and interiors of regular languages with subwords and superwords. *Theoretical Computer Science*, 610:91–107, 2016. doi:10.1016/j.tcs.2015.09.028.

**13** P. Karandikar and Ph. Schnoebelen. On the index of Simon's congruence for piecewise testability (v2), October 2013. This version available at arxiv:1310.1278v2. URL: http://arxiv.org/abs/1310.1278v2.

**14** P. Karandikar and Ph. Schnoebelen. Decidability in the logic of subsequences and supersequences. In *Proc. FST&TCS 2015*, volume 45 of *Leibniz International Proceedings in Informatics*, pages 84–97, 2015. doi:10.4230/LIPIcs.FSTTCS.2015.84.

**15** O. Klíma. Piecewise testable languages via combinatorics on words. *Discrete Mathematics*, 311(20):2124–2127, 2011. doi:10.1016/j.disc.2011.06.013.

**16** O. Klíma and L. Polák. Alternative automata characterization of piecewise testable languages. In *Proc. DLT 2013*, volume 7907 of *Lecture Notes in Computer Science*, pages 289–300. Springer, 2013. doi:10.1007/978-3-642-38771-5_26.

**17** L. Kontorovich, C. Cortes, and M. Mohri. Kernel methods for learning languages. *Theoretical Computer Science*, 405(3):223–236, 2008. doi:10.1016/j.tcs.2008.06.037.

**18** J. B. Kruskal. The theory of well-quasi-ordering: A frequently discovered concept. *Journal of Combinatorial Theory, Series A*, 13(3):297–305, 1972. doi:10.1016/0097-3165(72)90063-5.

**19** D. Kuske. Theories of orders on the set of words. *RAIRO Theoretical Informatics and Applications*, 40(1):53–74, 2006. doi:10.1051/ita:2005039.

**20** D. Kuske. Private communication, September 2015.

**21** T. Masopust. Piecewise testable languages and nondeterministic automata. arXiv:1603.00361 [cs.FL], March 2016. URL: http://arxiv.org/abs/1603.00361.

**22** T. Masopust and M. Thomazo. On the complexity of $k$-piecewise testability and the depth of automata. In *Proc. DLT 2015*, volume 9168 of *Lecture Notes in Computer Science*, pages 364–376. Springer, 2015. doi:10.1007/978-3-319-21500-6_29.

**23** O. Matz. On piecewise testable, starfree, and recognizable picture languages. In *Proc. FOSSACS '98*, volume 1378 of *Lecture Notes in Computer Science*, pages 203–210. Springer, 1998. doi:10.1007/BFb0053551.

**24** D. Perrin and J.-É. Pin. *Infinite words: Automata, Semigroups, Logic and Games*, volume 141 of *Pure and Applied Mathematics Series*. Elsevier Science, 2004.

**25** J.-É. Pin. *Varieties of Formal Languages*. Plenum, New-York, 1986.

**26** J. Rogers, J. Heinz, M. Fero, J. Hurst, D. Lambert, and S. Wibel. Cognitive and subregular complexity. In *Proc. FG 2012 & 2013*, volume 8036 of *Lecture Notes in Computer Science*, pages 90–108. Springer, 2013. doi:10.1007/978-3-642-39998-5_6.

**27** J. Sakarovitch and I. Simon. Subwords. In M. Lothaire, editor, *Combinatorics on Words*, volume 17 of *Encyclopedia of Mathematics and Its Applications*, chapter 6, pages 105–142. Cambridge Univ. Press, 1983.

**28** A. Salomaa, D. Wood, and Sheng Yu. On the state complexity of reversals of regular languages. *Theoretical Computer Science*, 320(2–3):315–329, 2004. doi:10.1016/j.tcs.2004.02.032.

**29** I. Simon. Piecewise testable events. In *Proc. 2nd GI Conf. on Automata Theory and Formal Languages*, volume 33 of *Lecture Notes in Computer Science*, pages 214–222. Springer, 1975. doi:10.1007/3-540-07407-4_23.

**30** I. Simon. Words distinguished by their subwords. In *Proc. WORDS 2003*, 2003.

**31** G. Zetzsche. An approach to computing downward closures. In *Proc. ICALP 2015*, volume 9135 of *Lecture Notes in Computer Science*, pages 440–451. Springer, 2015. doi:10.1007/978-3-662-47666-6_35.

## A Operations that do not preserve piecewise-testability

In this appendix we give examples showing that PT languages are not closed under many usual language-theoretic operations.

**Concatenation.** $a(a+b)^*$ is not PT: it contains $(ab)^k$ but not $b(ab)^k$, however $(ab)^k \sim_k b(ab)^k$ for any $k$. Hence the class PT is not closed under concatenation (even in the special case of prefixing with $a$) since $(a+b)^*$ is PT.

**Kleene star.** PTis not closed under Kleene star (recall that PTis a subvariety of the star-free languages): $aa$ is finite hence PT but $(aa)^*$ is not PT.

**Shuffle product.** $ab^*$ and $a^*$ are PT but their shuffle product $ab^* \sqcup\!\sqcup a^* = a(a+b)^*$ is not.

**Conjugacy.** Recall that the conjugates of $u$ are $\widetilde{u} \stackrel{\text{def}}{=} \{u_2u_1 \mid u = u_1u_2\}$, and we extend with $\widetilde{L} = \bigcup_{u\in L} \widetilde{u}$. Now $ac(a+b)^*$ is PT but $\widetilde{ac(a+b)^*} = (a+b)^*ac(a+b)^* + c(a+b)^*a$ is not.

**Renaming.** $c(a+b)^*$ is PT but applying the renaming $c \mapsto a$ yield $a(a+b)^*$.

**Erasing one letter.** This operation can be seen as the inverse of $L \mapsto L \sqcup\!\sqcup A$ where an arbitrary letter is inserted at an arbitrary position. Now $ac(a+b)^*$ is PT but erasing one letter yields $(a+c+ac)(a+b)^*$ which is not PT.

## B Image of PT-languages by inverse morphisms

For the sake of completeness, we give a proof of Eq. (4) from page 4:

▸ **Lemma.** *Let $\rho : A^* \to B^*$ be a monoid morphism. If $L \subseteq B^*$ is $n$-PT then $\rho^{-1}(L) \subseteq A^*$ is also $n$-PT.*

**Proof.** Consider a word $w \in B^*$ of length $p$ with $p \leqslant n$. We start by showing that $\rho^{-1}(\uparrow w)$ is $n$-PT. Clearly $\rho^{-1}(\uparrow w)$ is upward-closed and hence PT, it remains to be shown that all minimal elements of $\rho^{-1}(\uparrow w)$ are of length at most $n$. Let $v$ be a minimal element of $\rho^{-1}(\uparrow w)$. Then $w \sqsubseteq \rho(v)$. Write $w = w_1 \dots w_p$ with each $w_i$ a letter, and $v$ as $v_1 \dots v_q$ with each $v_j$ a letter. We have $w_1 \dots w_p \sqsubseteq \rho(v_1) \dots \rho(v_q)$. Further, by minimality of $v$, if any factor $\rho(v_j)$ is removed from the right hand side, the relation will no longer hold. Thus $q \leqslant p$ and so $q \leqslant n$. Since $\rho^{-1}(\uparrow w)$ is a union of sets of the form $\uparrow v$ with $|v| \leqslant n$, $\rho^{-1}(\uparrow w)$ is $n$-PT.

Finally, note that inverse morphisms preserve boolean operations, that is, $\rho^{-1}(B^* \setminus L) = A^* \setminus \rho^{-1}(L)$, and $\rho^{-1}(L_1 \cup L_2) = \rho^{-1}(L_1) \cup \rho^{-1}(L_2)$. Every $n$-PT language $L \subseteq B^*$ is a boolean combination of principal filters $\uparrow w$ with $|w| \leqslant n$, and so the result follows. ◂

## C A polynomial-time algorithm for the PT height of single words.

It is not too hard to compute the PT height of a singleton language as we now explain. For words $u, v \in A^*$, let $\delta(u,v) = \min\{n : u \not\sim_n v\}$ if $u \neq v$ and $\delta(u,v) = \infty$ if $u = v$.

Let us first describe a simple algorithm to compute $\delta(u,v)$, given $u$ and $v$. Assume $u \neq v$. Then $\delta(u,v)$ is the smallest length $n$ such that some word of length $n$ is a subword of exactly one of $u$ and $v$. For any word $w$, the canonical complete DFA $\mathcal{A}_{\downarrow w}$ for the set of all subwords of $w$ has $|w|+2$ states and is easy to build. Then $\delta(u,v)$ is the length of a shortest word in $L(\mathcal{A}_{\downarrow u}) \triangle L(\mathcal{A}_{\downarrow v})$, where $\triangle$ denotes symmetric difference. Using the product construction for DFAs, one can compute $\delta(u,v)$ in time $O(|u| \cdot |v| \cdot |A|)$. A more involved algorithm to compute $\delta(u,v)$ in time $O(|u| + |v| + |A|)$ is presented in [30].

Note that $h(u)$ is the smallest $n$ such that the equivalence class of $u$ under $\sim_n$ is just $\{u\}$. If $[u]_n$ is not a singleton, then it has infinitely many elements (see Lemma 2.1.4), in

particular, some word of length greater than $|u|$, and therefore some word $v$ such that $u \sqsubseteq v$ and $|v| = |u| + 1$ (see Lemma 2.1.2). The number of such words $v$ is at most $(|u| + 1) \cdot |A|$, and so computing $\delta(u, v)$ for all such $v$ allows us to compute $h(u)$:

$$h(u) = \max_{v \in u \sqcup\!\sqcup A} \delta(u, v)\,.$$

This gives an overall time complexity upper bound of $O(|u|^3 \cdot |A|^2)$. Using the algorithm from [30] to compute $\delta$, this can be improved to $O((|u| + |A|) \cdot |u| \cdot |A|)$.

## D Proof that $P_k$ and $N_k$ characterize $U_k$

▸ **Claim.** *For any $k \in \mathbb{N}$ and $u \in A_k^*$:*

$$\Big(\bigwedge_{v \in P_k} v \sqsubseteq u\Big) \wedge \Big(\bigwedge_{w \in N_k} w \not\sqsubseteq u\Big) \iff u = U_k. \tag{6}$$

**Proof.** By induction on $k$. For $k = 0$, $A_0$ is empty and there is only one word in $A_0^*$, namely $u = U_0 = \varepsilon$. It satisfies the positive constraint $U_0 \sqsubseteq u$ and there are no negative constraints in $N_0$.

Assume now that $k > 0$ and that the claim holds for $k - 1$. We prove the left-to-right implication: Since $P_k$ is not empty, the $P_k$ constraints $a_k^i v a_k^{\eta - i} \sqsubseteq u$ imply that $|u|_{a_k} \geqslant \eta$. However the $N_k$ constraint $a_k^{\eta+1} \not\sqsubseteq u$ implies that $u$ contains exactly $\eta$ occurrences of $a_k$ and can be written $u = v_0 a_k v_1 a_k \cdots a_k v_\eta$ with $v_i \in A_{k-1}^*$ for all $i = 0, \ldots, \eta$.

Consider some fixed $v_i$: for any $v \in P_{k-1}$ it holds that $v \sqsubseteq v_i$ since $a_k^i v a_k^{\eta - i} \sqsubseteq u$. Similarly $w \not\sqsubseteq v_i$ for any $w \in N_{k-1}$ since $a_k^i w a_k^{\eta - i} \not\sqsubseteq u$. The ind. hyp. now yields $v_i = U_{k-1}$, thus $u = U_{k-1} a_k U_{k-1} \cdots a_k U_{k-1} = U_k$. The right-to-left implication should now be clear and can be left to the reader. ◂

## E Computing $h(\downarrow U_k)$

Let $U_0 = \varepsilon$ and, for $k > 0$, $U_k = (U_{k-1} a_k)^\eta U_k$. Write $L_k$ for $|U_k|_{a_1}$ and note that $L_k = (\eta + 1) L_{k-1}$ when $k > 1$.

▸ **Claim.** *For any $k, r \in \mathbb{N}$, if $x \sqsubseteq U_k^r$, then $h(x) \leqslant 1 + r L_k$.*

**Proof.** By induction on $k$. For $k \leqslant 1$, $U_k^r = a_1^{r L_k}$ requires $x = a_1^\ell$ with $\ell \leqslant r L_k$ so $h(x) = 1 + \ell \leqslant 1 + r L_k$.

So assume $k > 1$. Let $m = |x|_{a_k}$ and factor $x$ as $x_0 a_k x_1 a_k \ldots a_k x_m$ so that $x_i \in A_{k-1}^*$ for all $i$. Now, for any $y \in A^*$, the following holds:

$$y = x \iff a_k^m \sqsubseteq y \wedge a_k^{m+1} \not\sqsubseteq y \wedge \bigwedge_{i=0}^{m} \bigwedge_{u \in A^{\leqslant h(x_i)}} (a_k^i u a_k^{m-i} \sqsubseteq y \iff u \sqsubseteq x_i)\,. \tag{12}$$

We deduce that $h(x) \leqslant \max(m+1, m+h(x_0), \ldots, m+h(x_m)) = m + \max(1, h(x_0), \ldots, h(x_m))$. Note that $x_i \sqsubseteq U_{k-1}^{r'}$ for $r' = r(\eta+1) - m$, so, by induction hypothesis, $h(x_i) \leqslant 1 + r' L_{k-1}$. Assuming $k > 1$, we thus have

$$\begin{aligned} h(x) \leqslant m + 1 + r' L_{k-1} &= m + 1 + [r(\eta+1) - m] L_{k-1} \\ &= 1 + m[1 - L_{k-1}] + r(\eta+1) L_{k-1} \leqslant 1 + r L_k\,. \end{aligned}$$

◂

▸ **Corollary 5.1.** *$h(\downarrow U_k^r) = 1 + r L_k$, and thus in particular, $h(\downarrow U_k) = 1 + L_k$.*

**Proof.** We use Eq. (5) and note that $a_1^{r L_k} \in \downarrow U_k^r$. Hence $h(\downarrow U_k^r) \geqslant h(a_1^{r L_k}) = 1 + r L_k$. ◂

## F Proofs for Lemmas 3.2 and 3.3

▸ **Lemma 3.2.** *If $u_1$ and $u_2$ are respectively $\ell_1$-rich and $\ell_2$-rich, then $v \sim_n v'$ implies $u_1vu_2 \sim_{\ell_1+n+\ell_2} u_1v'u_2$.*

**Proof.** A subword $x$ of $u_1vu_2$ can be decomposed as $x = x_1yx_2$ where $x_1$ is the longest prefix of $x$ that is a subword of $u_1$ and $x_2$ is the longest suffix of the remaining $x_1^{-1}x$ that is a subword of $u_2$. Thus $y \sqsubseteq v$ since $x \sqsubseteq u_1vu_2$. Now, since $u_1$ is $\ell_1$-rich, $|x_1| \geqslant \ell_1$ (unless $x$ is too short), and similarly $|x_2| \geqslant \ell_2$ (unless ...). Finally $|y| \leqslant n$ when $|x| \leqslant \ell_1 + n + \ell_2$, and then $y \sqsubseteq v'$ since $v \sim_n v'$, entailing $x \sqsubseteq u_1v'u_2$. A symmetrical reasoning shows that subwords of $u_1v'u_2$ of length $\leqslant \ell_1 + n + \ell_2$ are subwords of $u_1vu_2$ and we are done. ◂

▸ **Lemma 3.3.** *Consider two words $u, u'$ of richness $m$ and with rich factorizations $u = u_1a_1 \cdots u_ma_my$ and $u' = u'_1a_1 \cdots u'_ma_mv'$. Suppose that $v \sim_n v'$ and that $u_i \sim_{n+1} u'_i$ for all $i = 1, \ldots, m$. Then $u \sim_{n+m} u'$.*

**Proof.** By repeatedly using Lemma 3.2, one shows

$$\begin{aligned} u_1a_1u_2a_2 \cdots u_ma_mv &\sim_{n+m} u'_1a_1u_2a_2 \cdots u_ma_mv \\ &\sim_{n+m} u'_1a_1u'_2a_2 \cdots u_ma_mv \\ &\vdots \\ &\sim_{n+m} u'_1a_1u'_2a_2 \cdots u'_ma_mv \\ &\sim_{n+m} u'_1a_1u'_2a_2 \cdots u'_ma_mv' \,, \end{aligned}$$

using the fact that each factor $u_ia_i$ is rich. ◂

## G Proofs for Theorem 5.4

▸ **Lemma G.1.** *If $uLv \subseteq [uv]_n$, where $L \subseteq A^*$ is any language, then $u(\downarrow L)v \subseteq [uv]_n$.*

**Proof sketch.** Recall that $w_1 \sim_n w_2$ and $w_1 \sqsubseteq w_2$ implies $w_1 \sim_n w'$ for all $w_1 \sqsubseteq w' \sqsubseteq w_2$. ◂

▸ **Lemma 5.5** (from Section 5). *If $uB^*C^*B^*v \subseteq [uv]_n$, where $B, C \subseteq A$ are subalphabets, then $u(B \cup C)^*v \subseteq [uv]_n$.*

**Proof.** We prove that for any $m \in \mathbb{N}$, for any $w \in B^*(C^*B^*)^m$, for any $s \in A^{\leqslant n}$, $s \sqsubseteq uwv$ implies $s \sqsubseteq uv$. The proof is by induction on $m$, knowing that the claim holds by assumption for $m \leqslant 1$.

Assume therefore that $m \geqslant 2$ and write $w$ as $w = xyz$ with $x \in B^*C^*$, $y \in B^*(C^*B^*)^{m-2}$, and $z \in C^*B^*$. If $s \sqsubseteq uwv = uxyzv$ then $s$ can be factored as $s = s_us_xs_ys_zs_v$ with each factor $s_*$ a subword of the corresponding factor of $uwv$. Let $s' \stackrel{\text{def}}{=} s_us_xs_zs_v$ so that $s' \sqsubseteq uxzv$. Note that $xz \in B^*C^*B^*$ hence $s' \sqsubseteq uxzv$ entails $s' \sqsubseteq uv$ by assumption. Thus either $s_us_x \sqsubseteq u$ or $s_zs_v \sqsubseteq v$.

In the first case, $s = s_us_xs_ys_zs_v \sqsubseteq uyzv$ and since $yz \in B^*(C^*B^*)^{m-1}$ the induction hypothesis applies and yields $s \sqsubseteq uv$.

In the second case a symmetrical reasoning applies. ◂

▸ **Lemma G.2.** *If $uB^*C^*LD^*B^*v \subseteq [uv]_n$, where $B, C, D \subseteq A$ are subalphabets and $L \subseteq A^*$ is any language then $u(B \cup C)^*L(B \cup D)^*v \subseteq [uv]_n$.*

**Proof.** We assume that $L \neq \varnothing$ (otherwise the result holds trivially) so that $uB^*C^*LD^*B^*v \subseteq [uv]_n$ entails $uB^*C^*B^*v \subseteq [uv]_n$ (by Lemma G.1), hence $u(B \cup C)^*v \subseteq [uv]_n$ (by Lemma 5.5).

We now prove that for $s \in A^{\leqslant n}$ and $w \in (B^*C^*)^k L (D^*B^*)^\ell$, $s \sqsubseteq uwv$ implies $s \sqsubseteq uv$. The proof is by induction on $k + \ell \in \mathbb{N}$. Note that the Lemma's assumption handles all cases with $k \leqslant 1$ and $\ell \leqslant 1$.

Let us therefore assume $k > 1$ since the case where $\ell > 1$ is symmetrical. Assume $s \sqsubseteq uwv$ and write $w$ as $w = xyz$ with $x \in B^*C^*$, $y \in (B^*C^*)^{k-1}$, and $z \in L(D^*B^*)^\ell$.

Since $s \sqsubseteq uwv = uxyzv$ there is a factorization $s = s_u s_x s_y s_z s_v$ of $s$ with each factor $s_*$ embedding in the corresponding factor of $uxyzv$. Let now $s' \stackrel{\text{def}}{=} s_u s_x s_z s_v$: this word satisfies $s' \sqsubseteq uxzv$ and $|s'| \leqslant n$. Now $uxzv \in uB^*C^*L(D^*B^*)^\ell v$, so that we may apply the induction hypothesis and deduce $s' \sqsubseteq uv$ from $s' \sqsubseteq uxzv$. Thus either $s_u s_x \sqsubseteq u$ or $s_z s_v \sqsubseteq v$.

If $s_u s_x \sqsubseteq u$ we deduce

$$s = s_u s_x s_y s_z s_v \sqsubseteq uyzv. \tag{13}$$

Now $yz \in (B^*C^*)^{k-1} L (D^*B^*)^\ell$ so that we can apply the induction hypothesis and deduce $s \sqsubseteq uv$ from Eq. (13).

If $s_z s_v \sqsubseteq v$ we deduce

$$s = s_u s_x s_y s_z s_v \sqsubseteq uxyv. \tag{14}$$

Now $xy \in (B^*C^*)^k$ so that $uxyv \in [uv]_n$ as we observed at the beginning. Thus from Eq. (14) we deduce $s \sqsubseteq uv$. ◂

We now give an application of the above lemma in a form which we will use later:

▸ **Lemma 5.6** (from Section 5). *Suppose $L_1 B_1^* B_2^* \cdots B_\ell^* L_2 \subseteq [u]_n$ for some languages $L_1, L_2 \subseteq A^*$ and subalphabets $B_1, B_2, \ldots, B_\ell \subseteq A$ with $\ell \geqslant 3$. If $a \in B_1 \cap B_\ell$ then, letting $B'_i = B_i \cup \{a\}$, $L_1 B'^*_1 B'^*_2 \cdots B'^*_\ell L_2 \subseteq [u]_n$.*

**Proof.** By induction on $\ell$. Write $L_1 B_1^* B_2^* \cdots B_\ell^* L_2$ as

$$L_1 B_1^* \; a^* B_2^* \cdots B_{\ell-1}^* a^* \; B_\ell^* L_2 \,.$$

For every $u_1 \in L_1 B_1^*$ and $u_2 \in B_\ell^* L_2$, we have

$$u_1 \; a^* B_2^* \cdots B_{\ell-1}^* a^* u_2 \subseteq [u]_n \,.$$

Lemma G.2 gives $u_1 \; B'^*_2 B_3^* \cdots B_{\ell-2}^* B'^*_{\ell-1} \; u_2 \subseteq [u]_n$, hence $u_1 \; B'^*_2 B'^*_3 \cdots B'^*_{\ell-2} B'^*_{\ell-1} \; u_2 \subseteq [u]_n$ by the induction hypothesis. Since this applies to all $u_1 \in L_1 B_1^* = L_1 B'^*_1$ and $u_2 \in B_2^* L_2 = B'^*_2 L_2$, we have proven the lemma. ◂

▸ **Lemma 5.8** (from Section 5). *Let $L \subseteq A^*$ be $n$-PT. Let $k = |A|$ and $m = f_k(n)$. For every $u \in L$ there is a $D$-product $P_u$ of length $\ell \leqslant mk + m + k$ such that $u \in P_u \subseteq [u]_n \subseteq L$.*

In the above statement (and below) we abuse notation and let $P$ denote both a regular expression and the language (a subset of $A^*$) it denotes.

**Proof.** Assume $u \in L$. By the small-subword theorem, and since $L$ is closed under $\sim_n$, there exists a subword $v = a_1 \ldots a_\ell$ of $u$ with $v \sim_n u$ and $\ell = \ell \leqslant m$. Thus $u$ has the form

$$u = b_{0,1} \cdots b_{0,p_0} \, a_1 \, b_{1,1} \cdots b_{1,p_1} \, a_2 \cdots a_\ell \, b_{\ell,1} \cdots b_{\ell,p_\ell} \,.$$

Here the $b_{i,j}$'s are the letters from $u$ that do not occur in the subword $v$. To shorten notation, we write $u = \prod_{i=0}^{\ell}(a_i \prod_{j=1}^{p_i} b_{i,j})$, abusing notation by letting $a_0 = \varepsilon$.

By the pumping property (Lemma 2.1.3), we deduce that $P \stackrel{\text{def}}{=} \prod_{i=0}^{\ell}(a_i \prod_{j=1}^{p_i}\{b_{i,j}\}^*)$ is a D-product satisfying $u \in P \subseteq [u]_n \subseteq L$.

We now modify this product to take advantage of the more general pumping property. Let $P' = \prod_{i=0}^{\ell}(a_i \prod_{j=1}^{p_i} B_{i,j}^*)$ where $B_{i,j} = \{b_{i,j}\} \cup \{a \in A \mid \exists 1 \leqslant j_1 < j < j_2 \leqslant p_i \ : \ a = b_{i,j_1} = b_{i,j_2}\}$. That is, every subalphabet $\{b_{i,j}\}$ in $P$ is completed with any letter that appears both before and after $b_{i,j}$ in the same $i$-th segment $B_{i,1}^* \cdots B_{i,p_i}^*$. Now Lemma 5.6 ensures that $P \subseteq P' \subseteq [u]_n \subseteq L$ (and we still have $u \in P'$ since $P \subseteq P'$).

We now simplify $P'$ by repeatedly replacing a factor $B^*B'^*$ where $B \subseteq B'$ (or $B' \subseteq B$) by $B'^*$ (or $B^*$). This does not change the language denoted by $P'$. When no more simplifications are possible, we let $P_u \stackrel{\text{def}}{=} \prod_{i=0}^{\ell}(a_i \prod_{j=1}^{\ell_i} C_{i,j}^*)$ denote the simplified D-product. For any $i \in \{0, \ldots, \ell\}$, the sequence of sets $C_{i,1}, \ldots, C_{i,\ell_i}$ satisfies the hypothesis of Lemma 5.7, and thus $\ell_i \leqslant k = |A|$. This entails that $P_u$ has length bounded by $(m+1)(k+1)-1$ (recall that $a_0 = \varepsilon$), i.e. by $mk + m + k$. ◂

## H Proof that $w \sim_n w'$ for Lemma 6.2

Recall that $u = w_0a_1w_1w_2$, $v = w_0w_1a_2w_2$, with $w = w_0a_1w_1a_2w_2$ and $w' = w_0w_1a_2a_2w_2$. Since $w \sim_n v \sqsubseteq w'$, we only have to show that any subword of length at most $n$ of $w'$ is also a subword of $w$.

So let $s \sqsubseteq w'$ with $|s| \leqslant n$. Factorize $s$ as $s = v_0v_1s'v_2$ as follows:

1. Let $v_0$ be the longest prefix of $s$ such that $v_0 \sqsubseteq w_0$.
2. Having fixed $v_0$, let $v_1$ be the longest prefix of $(v_0)^{-1}s$ such that $v_1 \sqsubseteq w_1$.
3. Having fixed $v_0$ and $v_1$, let $v_2$ be the longest suffix of $(v_0v_1)^{-1}s$ such that $v_2 \sqsubseteq w_2$.

Then $s' \sqsubseteq a_2a_2$, since $s \sqsubseteq w'$. If $s' = \varepsilon$ or $s' = a_2$, then $s \sqsubseteq v \sqsubseteq w$, and we are done. So assume $s' = a_2a_2$. Let $t = v_0a_1v_1a_2v_2$. Then $t \sqsubseteq w$ and $|t| = |s| \leqslant n$, so $t$ is a subword of both $u$ and $v$.

▸ **Claim.** $v_1a_2v_2 \sqsubseteq a_1w_1w_2$.

**Proof.** The claim asserts that a certain suffix of $t$ is a subword of a certain suffix of $u$. We know that $t \sqsubseteq u$, i.e., $v_0a_1v_1a_2v_2 \sqsubseteq w_0a_1w_1w_2$. Hence if $v_1a_2v_2 \not\sqsubseteq a_1w_1w_2$, then $v_0a_1\alpha \sqsubseteq w_0$ for some nonempty prefix $\alpha$ of $v_1a_2v_2$. But this contradicts the definition of $v_0$. ◂

Now since $v_1a_2v_2 \sqsubseteq a_1w_1w_2$, we have $v_1a_2 \sqsubseteq a_1w_1$ or $a_2v_2 \sqsubseteq w_2$. Combining this with $v_1 \sqsubseteq w_1$ and $v_2 \sqsubseteq w_2$, we get $v_1a_2a_2v_2 \sqsubseteq a_1w_1a_2w_2$. Finally, this along with $v_0 \sqsubseteq w_0$ gives $s \sqsubseteq w$ as needed.