Logical Methods in Computer Science
Volume 15, Issue 2, 2019, pp. 6:1–6:27
https://lmcs.episciences.org/

Submitted Jun. 22, 2017
Published Apr. 30, 2019

# THE HEIGHT OF PIECEWISE-TESTABLE LANGUAGES AND THE COMPLEXITY OF THE LOGIC OF SUBWORDS

PRATEEK KARANDIKAR AND PHILIPPE SCHNOEBELEN

CMI, Chennai, India
*e-mail address*: prateek@cmi.ac.in

LSV, CNRS & ENS Paris-Saclay, France
*e-mail address*: phs@lsv.fr

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 article 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.

## 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 *piecewise-testable* if it is closed under $\sim_n$ for some $n \in \mathbb{N}$.

Piecewise-testable (PT) languages were introduced more than forty years ago in Imre Simon's doctoral thesis (see [Sim72, Sim75, SS83]) and have played an important role in the algebraic and logical theory of first-order definable languages, see [Pin86, DGK08, Klí11] and the references therein. They also constitute an important class of simple regular languages with applications in learning theory [KCM08], databases [BSS12], linguistics [RHF$^+$13], etc. The concept of PT languages has been extended to richer notions of "subwords" [Zet18], to trees [BSS12], infinite words [PP04, CP18], pictures [Mat98], or any combinatorial well-quasi-order [GS16].

*Key words and phrases:* Logic of subsequences; Piecewise-testable languages; Descriptive complexity.

An extended abstract of this work first appeared in the proceedings of the 25th EACSL Conference on Computer Science Logic (CSL 2016) [KS16].

Work partially supported by ANR grant ANR-14-CE28-0002 PACS. The first author is now with Google, London.

[1] Or "subsequences", not to be confused with "factors".


DOI:10.23638/LMCS-15(2:6)2019

When a language $L \subseteq A^*$ is PT, we further say that it is "$n$-PT" if it is closed under $\sim_n$, and the smallest such $n$ is called the PT *height* of $L$, denoted $h(L)$ in this article.

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 $L$ [DGK08]. In this article, 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, and were motivated by, open problems in the complexity analysis of a 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. [HM15]. Deciding whether a DFA or a NFA $\mathcal{A}$ recognises a $n$-PT language is coNP-complete or PSPACE-complete respectively (see [MT17] and the references therein). The methods underlying these algorithms usually provide a bound on $h(L)$ in terms of $\mathcal{A}$: Klíma and Polák showed that $h(L)$ is bounded by the maximal length of a simple path from an initial to a final state in $\mathcal{A}$ [KP13]. The currently best bounds on $h(L)$ based on automata for $L$ have been obtained by Masopust and Thomazo [MT15, Mas16].

When $L$ is obtained by operations on other languages, very little is known about PT heights. It is clear that $h(A^* \smallsetminus L) = h(L)$ 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.

*Our contribution.* We provide upper and lower 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 we 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 [KS15] 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 article.* Section 2 recalls the basic notions (subwords, Simon's congruence, etc.) and gives some first bounds relating PT heights and minimal 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 $\mathsf{FO}^2(A^*, \sqsubseteq)$. In passing, we characterise the expressive power of the $\mathsf{FO}^2(A^*, \sqsubseteq)$ logic, or equivalently of its quantifier-free fragment, via new notions of subword-recognizable and piecewise-testable relations on words.

## 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 letters that occur in $u$ is denoted by $\alpha(u)$. 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.

### 2.1. Subwords and superwords.

We say that a word $u$ is a *subword* (i.e., a subsequence) of $v$, or equivalently that $v$ is a *superword* of $u$, 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

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

(Formally, one should write $\uparrow_A u$ since the definition depends on the alphabet at hand, but we will leave $A$ implicit: it will always be clear from the context.) For example, $\downarrow ab = \{ab, a, b, \varepsilon\}$ and $\uparrow ab = A^* a A^* b A^*$.

This is generalised to the closures of whole languages, via $\uparrow L = \bigcup_{u \in L} \uparrow u$ and $\downarrow L = \bigcup_{u \in L} \downarrow u$. The Kuratowski closure axioms are satisfied:

$$\uparrow \varnothing = \varnothing\,, \quad L \subseteq \uparrow L = \uparrow\uparrow L\,, \quad \uparrow\Big(\bigcup_i L_i\Big) = \bigcup_i \uparrow L_i\,, \quad \uparrow\Big(\bigcap_i \uparrow L_i\Big) = \bigcap_i \uparrow L_i\,,$$

and similarly for downward closures. 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.

A variant of the closure operations is based on the strict ordering: we let

$$\uparrow_< u \stackrel{\text{def}}{=} \{v \mid u \sqsubset v\}\,, \quad \uparrow_< L \stackrel{\text{def}}{=} \bigcup_{u \in L} \uparrow_< u\,, \quad \downarrow_< u \stackrel{\text{def}}{=} \{v \mid v \sqsubset u\}\,, \quad \downarrow_< L \stackrel{\text{def}}{=} \bigcup_{u \in L} \downarrow_< u\,.$$

While these are not closure operations, the languages $\uparrow_< L$ and $\downarrow_< L$ are upward-closed and downward-closed, respectively. Since upward-closed and downward-closed languages are regular (Haines Theorem [Hai69], also a corollary of Higman's Lemma [Hig52]) we conclude that $\uparrow L$, $\downarrow L$, $\uparrow_< L$ 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 $L$. For example, $I(a^*b^*) = A^+$ and $I((aaa)^+) = A^* \setminus a^*$.

2.2. **Recognizable and rational relations over words.** Recall that a binary relation $R \subseteq A^* \times A^*$ is *rational* if it can be defined via an (asynchronous nondeterministic) transducer or, equivalently, via a regular expression using elements of $A^* \times A^*$, unions, concatenations and Kleene stars, see, e.g. [Ber79, Chap. 3] or [Sak09, Chap. 4]. It is well-known that, while $\mathsf{Rat}(A^* \times A^*)$ is (effectively) closed under composition —as well as union, concatenation, and Kleene star,— it is not closed under complement or intersection.

For example, we can define equality over $A^*$ as well as the subword relations (strict and non strict) via the following regular expressions:[2]

$$=_{A^*} = \left(\bigcup_{a\in A}\begin{bmatrix}a\\a\end{bmatrix}\right)^*, \quad \sqsubseteq_{A^*} = \left(\bigcup_{a\in A}\begin{bmatrix}a\\\varepsilon\end{bmatrix}\cup\begin{bmatrix}a\\a\end{bmatrix}\right)^*, \quad \sqsubset_{A^*} = \sqsubseteq_{A^*}\cdot\left(\bigcup_{a\in A}\begin{bmatrix}a\\\varepsilon\end{bmatrix}\right)\cdot\sqsubseteq_{A^*}. \tag{2.1}$$

Since $\sqsubseteq_{A^*}$ and $\sqsubset_{A^*}$ can even be defined by *deterministic* transducers, we deduce that their complements, $\not\sqsubseteq_{A^*}$ and $\not\sqsubset_{A^*}$, are rational. Finally, let us mention the following result:

**Proposition 2.1** [KS15]**.** *The incomparability relation $\perp_{A^*} \subseteq A^* \times A^*$ is rational. Consequently $I(L)$, i.e., the image of $L$ by $\perp_{A^*}$, is effectively regular for any regular $L$.*

We note that proving Proposition 2.1 cannot rely on the characterisation $\perp_{A^*} = \not\sqsubseteq_{A^*} \cap \not\sqsupseteq_{A^*}$ since the intersection of two rational relations is in general not rational, even when the two relations are given by deterministic transducers.

The rational relations over $A^*$ encompass the special case of the *recognizable relations*. Recall that $R \subseteq A^* \times A^*$ is recognizable —in the standard way, i.e., "by some morphism to a finite monoid"— if it is a finite union $R = L_1 \times L'_1 \cup \cdots \cup L_m \times L'_m$ of cartesian products where all $L_i$'s and $L'_i$'s are regular languages over $A$. Recognizable relations are rational but the converse does not hold, for example, the equality relation $=_{A^*}$ is not recognizable. We shall use the well-known and easy-to-see fact that $\mathsf{Rec}(A^* \times A^*)$ is (effectively) closed under boolean operations.

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

$$u \sim_n v \overset{\text{def}}{\iff} {\downarrow}u \cap A^{\leqslant n} = {\downarrow}v \cap A^{\leqslant n} . \tag{2.2}$$

In other words, $u \sim_n v$ if $u$ and $v$ have the same subwords of length at most $n$. For example $abab \sim_1 aabb$ (both words use the same letters) but $abab \not\sim_2 aabb$ ($ba$ is a subword of $abab$, not of $aabb$). Note that $u \sim_0 v$ for any $u, v$, and $u \sim_n u$ for any $n$. 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 [Sim75, SS83].

We further let

$$u \lesssim_n v \overset{\text{def}}{\iff} u \sim_n v \wedge u \sqsubseteq v . \tag{2.3}$$

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 $\sim_n$, introduced in [Sim72], is called Simon's congruence of order $n$.

[2] The expression for $\sqsubset_{A^*}$ uses the concatenation, denoted $R \cdot R'$, of relations, not their composition $R \circ R'$.

The following properties will be useful:

**Lemma 2.1.** *For all $u, v, v' \in A^*$ and $a, b \in A$:*

(1) *If $u \lesssim_n v$ then $u \sim_n w$ for all $w \in A^*$ such that $u \sqsubseteq w \sqsubseteq v$;*
(2) *When $n > 0$, $u \sim_n uv$ if, and only if, there exists a factorization $u = u_1 u_2 \cdots u_n$ such that $\alpha(u_1) \supseteq \alpha(u_2) \supseteq \cdots \supseteq \alpha(u_n) \supseteq \alpha(v)$;*
(3) *If $uav \sim_n ubv'$ and $a \neq b$ then $ubav \sim_n ubv'$ or $uabv' \sim_n uav$ (or both);*
(4) *If $u \sim_n v$ then there exists $w \in A^*$ such that $u \lesssim_n w$ and $v \lesssim_n w$;*
(5) *If $u \sim_n v$ and $|u| < |v|$ then there exists some $v'$ with $|v'| = |u|$ and such that $u \sim_n v' \sqsubseteq v$;*
(6) *If $uv \sim_n uav$ then $uv \sim_n ua^\ell v$ for all $\ell \in \mathbb{N}$;*
(7) *Every equivalence class of $\sim_n$ is a singleton or is infinite.*

*Proof.* (1) is by combining Eq. (2.2) with $\downarrow u \subseteq \downarrow w \subseteq \downarrow v$; (2–4) are Lemmas 3, 5, and 6 from [Sim75]; (5) is an immediate consequence of Theorem 4 from [Sim72, p. 91], showing that all minimal (wrt. $\sqsubseteq$) words in $[u]_n$ have the same length —see also Theorem 6.2.9 from [SS83]; (6) is in the proof of Corollary 2.8 from [SS83]; (7) follows from (1), (4) and (6). □

2.4. **Piecewise-testable languages.** A language $L \subseteq A^*$ is piecewise-testable (or PT) if it if closed under $\sim_n$ for some $n$ (and then we say that it is $n$-piecewise-testable, or $n$-PT). 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. Finally, we write $\mathsf{PT}$ for the class of piecewise-testable languages (over some alphabet $A$) and $\mathsf{PT}_n$ for the class of 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 hierarchy of varieties of regular languages.

**Fact 2.2** (Alternative characterisations of PT and $n$-PT languages)**.** *Let $L \subseteq A^*$. The following are equivalent:*

(1) *$L$ is $n$-PT (i.e., closed under $\sim_n$);*
(2) *$L$ is a finite union $[u_1]_n \cup [u_2]_n \cup \cdots \cup [u_m]_n$ of $\sim_n$ classes;*
(3) *$L$ is a finite 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 $\ell \leqslant n$;*
(4) *$L$ is definable in the $\mathcal{B}\Sigma_1[<]$ fragment[3] of first-order logic over words, via a formula involving only $n$ variables* [DGK08].

*The following are equivalent:*

(5) *$L$ is PT;*
(6) *$L$ is recognised by a finite and $\mathcal{J}$-trivial monoid* [Sim75, ST88, HP00, Klí11]*;*
(7) *$L$ is regular and its minimal DFA is partially ordered and satisfies the UMS property* [Sim75, Tra01]*;*
(8) *$L$ is regular and its minimal DFA is acyclic and locally confluent* [KP13].

The characterisations (3), (4), (7) and (8) are useful for showing that a language is PT —or even $n$-PT in the case of (3) and (4)—. For example, with alphabet $A = \{a, b, c\}$, the language $a^+b^*$ can be defined via required and excluded minors, as in:

$$u \in a^+b^* \iff a \sqsubseteq u \wedge ba \not\sqsubseteq u \wedge c \not\sqsubseteq u\,. \tag{2.4}$$

[3]That is, the boolean closure of the existential fragment.

This definition of $a^+b^*$ directly translates into a $\mathcal{B}\Sigma_1$ formula, or into a finite boolean combination of filters:

$$a^+b^* = \uparrow a \smallsetminus \uparrow ba \smallsetminus \uparrow c = A^*aA^* \smallsetminus A^*bA^*aA^* \smallsetminus A^*cA^* \,. \tag{2.5}$$

Furthermore, since the minors in Eq. (2.4) have length at most 2 (they are $a$, $ba$, and $c$), we conclude that $h(a^+b^*) \leqslant 2$.

The characterisations (7) and (8) are also useful for showing that a language is not PT (see examples in Section 2.6 below). Finally (1) is very useful for showing that a language is not $n$-PT for a given $n$: 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., one sees that $a^+b^*$ is not 1-PT since $ab \sim_1 ba$ while only $ab$ is in $a^+b^*$. We may now conclude that $h(a^+b^*) = 2$.

Some examples of (families of) PT languages are:

**All finite languages:** $u \sim_n v$ and $n > |u|$ imply $u = v$. Thus $[u]_n = \{u\}$ and any $L = \{u_1, \ldots, u_m\} \subseteq A^{<n}$ can be expressed as $L = [u_1]_n \cup \cdots \cup [u_m]_n$. By characterisation (2), $L$ is PT.

**All upward-closed languages:** By Haines Theorem [Hai69, Hig52], any language $L \subseteq A^*$ has finitely many minimal elements (wrt $\sqsubseteq$), i.e., $\min(L)$ is some $\{u_1, \ldots, u_m\}$. This entails $\uparrow L = \uparrow u_1 \cup \cdots \cup \uparrow u_m$, which is PT by characterisation (3).

**All downward-closed languages:** They are the complements of upward-closed languages, hence PT again by characterisation (3).

In sections 3, 4 and 5 respectively, we analyse the PT heights of languages belonging to the above three families.

2.5. **Piecewise-testable relations.** Following the generic pattern laid out in [GS16], we say that a relation $R \subseteq A^* \times A^*$ is *piecewise-testable* if it is a finite boolean combination of principal filters $\uparrow(u, v)$ in the product ordering $(A^* \times A^*, \sqsubseteq \times \sqsubseteq)$. The relation is $n$-PT if the boolean combination only uses filters $\uparrow(u, v)$ with $|(u, v)| \leqslant n$, where we define $|(u, v)| \stackrel{\text{def}}{=} \max(|u|, |v|)$.

Since $\uparrow(u, v) = (\uparrow u) \times (\uparrow v)$, we see that piecewise-testable relations are recognizable. Using

$$\begin{aligned} (A^* \times A^*) \smallsetminus \uparrow(u, v) \;&=\; (A^* \smallsetminus \uparrow u) \times A^* \;\cup\; A^* \times (A^* \smallsetminus \uparrow v) \\ &=\; (\neg\uparrow u) \times \uparrow\varepsilon \;\cup\; \uparrow\varepsilon \times (\neg\uparrow v) \end{aligned}$$

and de Morgan's laws, we further see that a finite boolean combination of filters $\uparrow(u, v)$ can be written as some $\bigcup_i \bigcap_j (\pm\uparrow u_{i,j}) \times (\pm\uparrow v_{i,j})$, i.e., $\bigcup_i \big((\bigcap_j \pm\uparrow u_{i,j}) \times (\bigcap_j \pm\uparrow v_{i,j})\big)$. Finally, any $n$-PT relation can be written under the form $R = L_1 \times L'_1 \cup \cdots \cup L_m \times L'_m$ where all $L_i$'s and $L'_i$'s are $n$-PT languages. Hence PT relations form a subclass, denoted $\mathsf{PT}(A^* \times A^*)$ of $\mathsf{Rec}(A^* \times A^*)$. We shall not use PT relations until Section 7 and, for the moment, keep focused on PT languages.

2.6. **Closure properties of PT languages.** By definition (see Fact 2.2), any class $\mathsf{PT}_n$ is closed under boolean operations. Furthermore, $\mathsf{PT}_n$ is also closed under (left and right) quotients and under inverse morphisms [Thé81]. In terms of PT height, the above statements can be written as

$$h\Big(\bigcup_i L_i\Big) \leqslant \max\,\{h(L_i)\}_i\,, \quad h\Big(\bigcap_i L_i\Big) \leqslant \max\,\{h(L_i)\}_i\,, \qquad h(\neg L) = h(L)\,, \tag{2.6}$$

$$h(\rho^{-1}(L)) \leqslant h(L) \text{ for } \rho : A^* \to B^* \text{ a morphism}, \qquad h(u^{-1}L\,v^{-1}) \leqslant h(L)\,. \tag{2.7}$$

Note that we can allow arbitrary unions and intersections in Eq. (2.6) since, for fixed $A$, there are only finitely many languages in $\mathsf{PT}_n$.

Let us also mention that $\mathsf{PT}_n$ is closed under taking mirror images: writing $\overleftarrow{a_1 a_2 \cdots a_\ell}$ for the mirror word $a_\ell \cdots a_2 a_1$, and letting $\overleftarrow{L} = \{\overleftarrow{u} \mid u \in L\}$ for a language $L \subseteq A^*$, one has $h(\overleftarrow{L}) = h(L)$.

Beyond that, $\mathsf{PT}$ is not closed under any of the usual language-theoretic operations as we now illustrate.

**Concatenation and prefixing:** $a(a+b)^*$ is not PT: it is not closed under any $\sim_k$ since it contains $(ab)^k$ but not $b(ab)^k$ while $(ab)^k \sim_k b(ab)^k$.[4] Since $(a+b)^*$ is PT, we see that the class $\mathsf{PT}$ is not closed under concatenation, even in the special case of prefixing with $a$.

**Kleene star:** $\mathsf{PT}$ is not closed under Kleene star (recall that $\mathsf{PT}$ is a subvariety of the star-free languages): $aa$ is finite hence PT but $(aa)^*$ is not PT since its minimal automaton is not acyclic.

**Shuffle product:** $ab^*$ and $a^*$ are PT but their shuffle product $ab^* ⧢ a^* = a(a+b)^*$ is not as just shown, see [HS19] for PT shufflings.

**Conjugacy:** Recall that the conjugates of $u$ are $\widetilde{u} \stackrel{\text{def}}{=} \{u_2 u_1 \mid u = u_1 u_2\}$, and we extend with $\widetilde{L} = \bigcup_{u \in L} \widetilde{u}$. Now $L = ac(a+b)^*$ is PT but $\widetilde{L} = (a+b)^* ac(a+b)^* + c(a+b)^* a$ is not.

**Renaming:** $c(a+b)^*$ is PT but applying the renaming $c \mapsto a$ yield $a(a+b)^*$ which is not.

**Erasing one letter:** This operation can be seen as the inverse of $L \mapsto L ⧢ A$ 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.

Finally, we are only aware of one more positive instance of a closure property, "$I(L)$ is PT when $L$ is", but proving this is the topic of Section 6.

2.7. **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 minimal DFA for $L$ [Yu05].

The bounds given in Eqs. (2.6) and (2.7) let us contrast the height of a PT language with its state complexity. If $L$ is a PT language, one has $h(L) \leqslant sc(L)$ (equality occurs e.g. when $L = \{a^\ell\}$) since $sc(L)$ bounds the depth of the minimal automaton for $L$, i.e., the maximum length of a simple path from the initial to some final state, which in turns bounds $h(L)$ [KP13, MT15].

In the other direction, we can prove

[4] An alternative proof is by observing that the minimal automaton for $(a+b)^*a$ —the mirror language— is not acyclic.

**Theorem 2.3.** *Let $A$ be an alphabet of size $k$ with $k > 1$. Suppose $L \subseteq A^*$ is $n$-PT. Then the minimal DFA for $L$ has at most $m$ states,*[5] *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 $2^{O(n^{k-1}\log n)}$, where $n = h(L)$.*

*Proof.* Write $L$ under the form $L = [u_1]_n \cup \cdots \cup [u_m]_n$ and consider the *$\sim_n$-canonical DFA*, i.e., the DFA $\mathcal{A} = (Q, A, i, ., F)$ whose states are all classes $[w]_n$ for $w \in A^*$ and with transitions given by $[w]_n.a = [wa]_n$. With initial state $i = [\varepsilon]_n$, this automaton reaches $[w]_n$ upon reading $w$. With accepting states $F = \{[u_1]_n, \ldots, [u_m]_n\}$, it recognises exactly $L$. In [KKS15] we showed that the number of equivalence classes of $\sim_n$, i.e., $|Q|$, is bounded by $m$. This in turns bound $sc(L)$. □

The general situation is that $h(L)$ can be much smaller than $sc(L)$ as we shall see in the following sections. More importantly, PT height is a more robust measure than state complexity. For example, state complexity can increase exponentially when building boolean combinations of regular languages (while PT height does not increase). This classic phenomenon occurs even when we restrict to PT languages, and even if we use nondeterministic state complexity, see the examples in [KNS16]. This is not limited to boolean operations: for example, we saw that $h\big(\overleftarrow{L}\big) = h(L)$ but $sc\big(\overleftarrow{L}\big)$ cannot be bounded by a polynomial of $sc(L)$, even in the case of finite, hence PT, languages [SWY04].

## 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 $\{u\} = {\uparrow}u \smallsetminus \bigcup_{v\in\{u\}\sqcup A} {\uparrow}v$, which entails $h(\{u\}) \leqslant |u| + 1$. Here we used a *shuffle product* notation, $\{u\} \sqcup A$, to denote $\{v\ :\ u \sqsubseteq v \wedge |v| = |u|+1\}$, i.e., the set of all words obtained from $u$ by inserting, at some position, one letter from $A$. Below we often omit set-theoretical parentheses when denoting singletons, writing e.g. "$h(u)$" or "$u \sqcup A$".

The $|u| + 1$ upper bound for $h(u)$ is tight. For example,

$$h(a^\ell) = \ell + 1\ . \tag{3.1}$$

(To see that $h(a^\ell) > \ell$, one notes that $a^\ell \sim_\ell a^{\ell+1}$.) However, words on more than one letter can generally be described within some PT height lower than their length. For example

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

showing $h(aabb) \leqslant 3$. (Note that $h(aabb) > 2$ since $aabbb \sim_2 aabb$, thus $h(aabb) = 3$.)

It turns out that the PT height of words can be much lower than their length as we shall see in section 3.1, but before considering lower bounds on $h(u)$, let us make some easier observations.

**Proposition 3.1.** *The PT-height of a word can be computed in polynomial time.*

[5] It is shown in [MT15, MT17] that the depth (not the size) of the minimal DFA is bounded by $\binom{n+k}{n} - 1$.

*Proof.* Following [SS83], we let $\delta(v, v') \stackrel{\text{def}}{=} \max\{n \mid v \sim_n v'\}$ for any two words $v, v' \in A^*$. We now claim that

$$h(u) = 1 + \max\{\delta(u, v) \mid v \in u \sqcup\!\sqcup A\}\,. \tag{3.2}$$

To prove the claim, we note that $u \neq v$ entails $\delta(u, v) < h(u)$ since, by definition of $h(u)$, $u \sim_{h(u)} v$ entails $u = v$. In the other direction, let $n = h(u) - 1$ so that $[u]_n$ is not a singleton. Then $[u]_n$ is infinite (Lemma 2.1 (7)) and in particular contains some word $w$ with $u \sqsubset w$ (Lemma 2.1 (4)). In fact $[u]_n$ contains all words between $u$ and $w$ (Lemma 2.1 (1)) hence some $w' \in u \sqcup\!\sqcup A$. Thus the right-hand side of (3.2) is at least $1 + \delta(u, w')$, i.e., at least $n + 1 = h(u)$.

We have thus reduced the computation of $h(u)$ to polynomially many $\delta(u, v)$ computations and now rely on the fact that $\delta$ can be computed in polynomial time [Sim03].[6] □

Proposition 3.1 can be used to compute the PT height of finite languages in polynomial time: for such languages, the inequality in Eq. (2.6) becomes

$$h(\{u_1, \ldots, u_m\}) = \max\{h(u_1), \ldots, h(u_m)\}\,. \tag{3.3}$$

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 (7). Hence $[u_i]_n = \{u_i\}$ and $h(u_i) \leqslant n$.

### 3.1. **Words with low PT height.**

We 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 $\lambda \in \mathbb{N}$. We let $U_0 \stackrel{\text{def}}{=} \varepsilon$ and, for $k > 0$, $U_k \stackrel{\text{def}}{=} (U_{k-1}a_k)^\lambda U_{k-1}$. For example, for $\lambda = 3$ and $k = 2$, one has $U_2 = a_1^3 a_2 a_1^3 a_2 a_1^3 a_2 a_1^3 = a_1a_1a_1a_2a_1a_1a_1a_2a_1a_1a_1a_2a_1a_1a_1$.

The rest of Section 3.1 establishes the following bounds for any $k, \lambda \in \mathbb{N}$:

$$|U_k| = (\lambda + 1)^k - 1\,, \qquad h(U_k) = k\lambda + 1\,, \qquad h(\downarrow U_k) = \lambda(\lambda + 1)^{k-1} + 1\,. \tag{3.4}$$

The first equality, $|U_k| = (\lambda + 1)^k - 1$, is easily seen by induction on $k$.

To show $h(U_k) = k\lambda + 1$ we use some auxiliary languages $P_k, N_k \subseteq A_k^*$ defined inductively by the following expressions:

$$P_0 \stackrel{\text{def}}{=} \{\varepsilon\}, \qquad\qquad N_0 \stackrel{\text{def}}{=} \varnothing, \tag{3.5}$$

and, for $k > 0$,

$$P_k \stackrel{\text{def}}{=} \sum_{i=0}^{\lambda} a_k^i \cdot P_{k-1} \cdot a_k^{\lambda - i}, \qquad\qquad N_k \stackrel{\text{def}}{=} a_k^{\lambda+1} + \sum_{i=0}^{\lambda} a_k^i \cdot N_{k-1} \cdot a_k^{\lambda - i}. \tag{3.6}$$

The words in $P_k$ and $N_k$ are used as positive, and respectively negative, constraints in the following claim.

*Claim* 3.2. For any $k \in \mathbb{N}$ and $u \in A_k^*$:

$$\Big(\bigwedge_{v \in P_k} v \sqsubseteq u\Big) \;\wedge\; \Big(\bigwedge_{w \in N_k} w \not\sqsubseteq u\Big) \iff u = U_k. \tag{3.7}$$

[6] It is also possible to directly compute $h(u)$ in time and space $O(|h| \cdot |A|)$ by adapting the techniques used in [FK18], where the goal is to compute a canonical $\sim_n$-equivalent for $u$.

*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 $\varepsilon \sqsubseteq u$ (from $P_0$) 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 $a_k^i v a_k^{\lambda-i} \sqsubseteq u$ imply that $|u|_{a_k} \geqslant \lambda$. However the $N_k$ constraint $a_k^{\lambda+1} \not\sqsubseteq u$ implies that $u$ contains exactly $\lambda$ occurrences of $a_k$ and can be written $u = v_0 a_k v_1 a_k \cdots a_k v_\lambda$ with $v_i \in A_{k-1}^*$ for all $i = 0, \ldots, \lambda$.

Consider some fixed $v_i$: for any $v \in P_{k-1}$ it holds that $v \sqsubseteq v_i$ since $a_k^i v a_k^{\lambda-i} \sqsubseteq u$. Similarly $w \not\sqsubseteq v_i$ for any $w \in N_{k-1}$ since $a_k^i w a_k^{\lambda-i} \not\sqsubseteq u$. The induction hypothesis 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. □

Claim 3.2 entails $h(U_k) \leqslant k\lambda + 1$ since the words in $P_k$ have length $k\lambda$ and the words in $N_k$ have length at most $k\lambda + 1$.

It remains to show that $h(U_k) > k\lambda$, i.e., that $\{U_k\}$ is not closed under $\sim_{k\lambda}$. For this we factor $U_k$ under the form

$$U_k = (U_{k-1}a_k)^\lambda (U_{k-2}a_{k-1})^\lambda (U_{k-3}a_{k-2})^\lambda \ldots (U_0 a_1)^\lambda \, . \tag{3.8}$$

Using Lemma 2.1 (2), this factorization in $k\lambda$ factors involving decreasing alphabets proves $U_k \sim_{k\lambda} U_k a_1$ and concludes the proof of the second equality in Eq. (3.4).

To prove the third equality in Eq. (3.4), we write $L_k$ for $|U_k|_{a_1}$ and note that $L_k = (\lambda+1)L_{k-1}$ when $k > 1$.

*Claim* 3.3. For any $k, r \in \mathbb{N}$ and $u \in A_k^*$, if $u \sqsubseteq U_k^r$, then $h(u) \leqslant 1 + rL_k$.

*Proof.* By induction on $k$. For $k \leqslant 1$, $u \sqsubseteq U_k^r = a_1^{rL_k}$ requires $u = a_1^\ell$ with $\ell \leqslant rL_k$. Eq. (3.1) then gives $h(u) = 1 + \ell \leqslant 1 + rL_k$.

So assume $k > 1$. Let $m = |u|_{a_k}$ and factor $u$ as $u_0 a_k u_1 a_k \ldots a_k u_m$ so that $u_i \in A_{k-1}^*$ for all $i = 0, \ldots, m$. We then derive a PT-characterisation of $u$ from PT-characterisations of the $u_i$'s: $u$ is the only word in $A^*$ that satisfies

$$a_k^m \sqsubseteq u \wedge a_k^{m+1} \not\sqsubseteq u \wedge \bigwedge_{i=0}^{m} \bigwedge_{w \in A_{k-1}^{\leqslant h(u_i)}} (a_k^i w a_k^{m-i} \sqsubseteq u \iff w \sqsubseteq u_i) \, . \tag{3.9}$$

We deduce that $h(u) \leqslant \max(m+1, m+h(u_0), \ldots, m+h(u_m)) = m + \max(1, h(u_0), \ldots, h(u_m))$.

Now recall that $u$ has $m$ occurrences of $a_k$ while $U_k^r$ has $r(\lambda+1)$. This implies that any $u_i$ in the decomposition of $u$ is a subword of $U_{k-1}^{r'}$ for $r' = r(\lambda+1) - m$, so, by induction hypothesis, $h(u_i) \leqslant 1 + r'L_{k-1}$. Assuming $k > 1$, we thus have

$$\begin{aligned} h(u) \leqslant m + 1 + r'L_{k-1} &= m + 1 + [r(\lambda+1) - m]L_{k-1} \\ &= 1 + m[1 - L_{k-1}] + r(\lambda+1)L_{k-1} \leqslant 1 + rL_k \, , \end{aligned}$$

establishing the claim. □

**Corollary 3.4.** *$h(\downarrow U_k^r) = 1 + rL_k$, and thus in particular, $h(\downarrow U_k) = 1 + L_k$.*

*Proof.* Since $\downarrow U_k^r$ is finite, Claim 3.3 and Eq. (3.3) entail $h(\downarrow U_k^r) \leqslant 1 + rL_k$. On the other hand, $a_1^{rL_k} \in \downarrow U_k^r$. Hence $h(\downarrow U_k^r) \geqslant h(a_1^{rL_k}) = 1 + rL_k$. □

3.2. **Rich words and rich factorizations.** Assume a fixed $k$-letter alphabet $A$. We say that a word $u$ is *rich* if $\alpha(u) = A$, i.e., the $k$ letters of $A$ all occur in $u$, 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 $u = r_1 \cdots r_\ell u'$ where the $\ell$ factors $r_1, \ldots, r_\ell$ 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 3.5.** *If $u_1$ and $u_2$ are respectively $\ell_1$-rich and $\ell_2$-rich, then $v \sim_n v'$ implies $u_1 v u_2 \sim_{\ell_1+n+\ell_2} u_1 v' u_2$.*

*Proof.* A subword $x$ of $u_1 v u_2$ can be decomposed as $x = x_1 y x_2$ 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 $x \sqsubseteq u_1 v u_2$. Now, since $u_1$ is $\ell_1$-rich, we have $|x_1| \geqslant \min(\ell_1, |x|)$, and similarly $|x_2| \geqslant \min(\ell_2, |x_1^{-1}x|)$. Finally $|y| \leqslant n$ when $|x| \leqslant \ell_1 + n + \ell_2$, and then $y \sqsubseteq v'$ since $v \sim_n v'$, entailing $x \sqsubseteq u_1 v' u_2$. A symmetrical reasoning shows that subwords of $u_1 v' u_2$ of length $\leqslant \ell_1 + n + \ell_2$ are subwords of $u_1 v u_2$ and we are done. $\square$

The *rich factorization* of $u \in A^*$ is the decomposition $u = u_1 a_1 \cdots u_m a_m v$ defined by induction in the following way: if $u$ is poor, we let $m = 0$ and $v = u$; otherwise $u$ is rich, we let $u_1 a_1$ (with $a_1 \in A$) be the shortest prefix of $u$ that is rich and let $u_2 a_2 \cdots u_m a_m v$ be the rich factorization of the remaining suffix $(u_1 a_1)^{-1} u$. 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 3.6.** *Consider two words $u, u'$ of richness $m$ and with rich factorizations $u = u_1 a_1 \cdots u_m a_m v$ and $u' = u'_1 a_1 \cdots u'_m a_m v'$. 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.* Since each factor $u_i a_i$ is rich, one gets

$$\begin{aligned} u_1 a_1 u_2 a_2 \cdots u_m a_m v \ &\sim_{n+m}\ u'_1 a_1 u_2 a_2 \cdots u_m a_m v \ \sim_{n+m}\ u'_1 a_1 u'_2 a_2 \cdots u_m a_m v \\ &\sim_{n+m}\ \cdots\ \sim_{n+m}\ u'_1 a_1 u'_2 a_2 \cdots u'_m a_m v \ \sim_{n+m}\ u'_1 a_1 u'_2 a_2 \cdots u'_m a_m v' \,, \end{aligned}$$

by repeated uses of Lemma 3.5. $\square$

3.3. **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 article.

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

$$f_1(n) = n \,, \tag{3.10}$$

$$f_{k+1}(n) = \max_{0 \leqslant m \leqslant n} \big( m f_k(n+1-m) + m + f_k(n-m) \big) \,. \tag{3.11}$$

In the rest of the article, we shall simplify statements involving the $f_k(n)$ bound by relying on the following bound (proved in the Appendix):

$$f_k(n) \leqslant \Big( \frac{n+2k-1}{k} \Big)^k - 1 < \Big( \frac{n}{k} + 2 \Big)^k \,. \tag{3.12}$$

**Theorem 3.7** (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 $n \leqslant \min(\ell, |u|)$. So taking $v = a^\ell$ for $\ell = \min(n, |u|)$ proves the claim.

When $k > 1$ we consider the rich factorization $u = u_1 a_1 u_2 a_2 \cdots u_m a_m u'$ 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 $u_i a_i \sim_{n'} v_i a_i$. 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 $v = v_1 a_1 \cdots v_m a_m v'$, so that $v \sqsubseteq u$ and $|v| \leqslant m f_{k-1}(n') + m + f_{k-1}(n' - 1)$. We deduce $|v| \leqslant f_k(n)$ using Eq. (3.11) and since $n' = n + 1 - m$. That $v \sim_n u$, hence $v \lesssim_n u$, is an application of Lemma 3.6: $v_1 a_1 \cdots v_m a_m v'$ 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 3.7 does not depend on $u$.

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

**Proposition 3.8** (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 $k(\ell + 1)^{1/k} - 2k + 1 \leqslant h(L) \leqslant \ell + 1$.*

*Proof.* Thanks to Eq. (3.3), 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. (3.12) one gets $\ell \leqslant f_k(n) \leqslant \left(\frac{n+2k-1}{k}\right)^k - 1$. This gives $n \geqslant k(\ell + 1)^{1/k} - 2k + 1$ as announced. The upper bound $h(L) \leqslant \ell + 1$ was observed earlier. □

*Remark* 3.9 (On Tightness)*.* We already noted that the $\ell + 1$ upper bound is tight. The lower bound is quite good: for $U_k$ seen above, $\ell = (\lambda + 1)^k - 1$, so that $\ell \leqslant \left(\frac{n+2k-1}{k}\right)^k - 1$ gives $n = h(U_k) \geqslant k\lambda - k + 1$ while we know $h(U_k) = k\lambda + 1$. □

Finding tight bounds for the trade-off between word length and PT-height is an interesting open problem. The existing gap in Proposition 3.8 can be narrowed at one end by improving the small-subword theorem and, at the other end, by discovering words with small PT-height as a function of their length. In this direction, we note that our $U_k$ words provably do not hold the record: for example, for $k = 3$ and $w = a^3 b^4 a^2 c^4 a^4 c b^3 c^3 b^2 =$ *aaabbbbaaccccaaaacbbbcccbb*, we have $|w| = 26$ and $h(w) = 6$, to be compared with $|U_3| = 26$ and $h(U_3) = 7$ when $\lambda = 2$.

## 4. Upward closures

Recall that $\uparrow L$ is PT for any $L \subseteq A^*$. 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 these languages as a function of $L$.

We first note that, in the special case where $L$ is a singleton, the PT height of $\uparrow L$ and $I(L)$ always coincide with word length:[7]

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

$$h(\uparrow u) = |u|\,, \qquad\qquad h(I(u)) = h(\uparrow u \cup \downarrow u) = \begin{cases} |u| & \text{if } |A| \geqslant 2, \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

*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 each 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 $u \not\sqsubseteq v$ and $u \sqsubseteq v'$ hence $\uparrow u$ is not closed under $\sim_{\ell-1}$.

Now for $I(u)$, we note that $h(I(u)) \leqslant \max(h(\uparrow u), h(\downarrow_< u))$ since $I(u) = A^* \smallsetminus (\uparrow u \cup \downarrow_< u)$, and that $\max(h(\uparrow u), h(\downarrow_< u)) = \ell$ since $h(\uparrow u) = \ell$ and 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 $A^*$ so $h(I(u)) > 0$. ☐

**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 $h(\uparrow L) \leqslant m$ and $h(\uparrow_< L) \leqslant m + 1$.*

*Proof.* Since $\uparrow L = \uparrow \min(L)$ and since $\min(L)$ is finite (by Higman's Lemma), we have $h(\uparrow L) = h(\bigcup_{u \in \min(L)} \uparrow u) \leqslant \max_{u \in \min(L)} h(\uparrow u) = \max_{u \in \min(L)} |u| \leqslant m$.

Now since $\uparrow_< L = (\uparrow L) \smallsetminus \min(L)$, we deduce $h(\uparrow_< L) \leqslant \max(h(\uparrow L), h(\min(L)))$. But $h(\min(L)) \leqslant m + 1$ by Proposition 3.8. ☐

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 $h(\uparrow L) \leqslant m$ while $h(\uparrow_< L) \leqslant m + 1$ and $h(\min(L)) \leqslant m + 1$.*
(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.* (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 smallest 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 bounding $h(\uparrow L)$ in terms of $h(L)$, assuming that $L$ is PT.

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

**Theorem 4.4** (Upward closures of PT languages)**.** *Suppose that $L \subseteq A^*$ is PT and let $k = |A|$ and $m = f_k(h(L))$. Then*

$$h(\uparrow L) \leqslant m\,, \qquad h(\uparrow_< L) \leqslant m+1\,, \qquad h(\min(L)) \leqslant m+1\,.$$

*Proof.* By the small-subword theorem, and since $L$ is closed under $\sim_{h(L)}$, 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 $k, \lambda \geqslant 1$, the language $L = \{U_k\}$ has $h(L) = h(U_k) = k\lambda + 1$ so that Theorem 4.4 with Eq. (3.12) give $h(\uparrow U_k) \leqslant f_k(k\lambda+1) \leqslant (\lambda+2)^k - 1$. On the other hand we know that $h(\uparrow U_k) = (\lambda+1)^k - 1$ 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.

**Definition 5.1.** A *D-product over $A$* 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 $P$.

As is common, we abuse notation and let $P$ denote both a regular expression and the associated language.

We note that our D-products are slightly more general than the monomials of the form $B_0^* a_1 B_1^* a_2 \cdots a_n B_n^*$ considered in [DGK08], where a strict alternation is imposed between star factors and letter factors. However, any D-product is easily translated as a polynomial (a finite sum of monomials) by replacing any two consecutive letter factors $a \cdot a'$ by the equivalent $a \cdot \varnothing^* \cdot a'$, and any two consecutive star factors $B^* \cdot B'^*$ by $B^* + \sum_{a \in B'} \big(B^* \cdot a \cdot B'^*\big)$ and then distributing concatenations over unions. Thus the languages described by finite unions of D-products are exactly the languages described by polynomials (see [DGK08, PW97] for algebraic and logical characterisations).

D-products generalise words, and they share with words their nice upper bound on the PT height of downward closures:

**Proposition 5.2.** *Let $P$ be a D-product of length $\ell$. Then $h(\downarrow P) \leqslant \ell+1$ 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 $P' = \downarrow P$. We claim that any residual $w^{-1}P''$ of a suffix $P''$ of $P'$ is either the empty language $\varnothing$, or is itself a suffix of $P'$. The claim is proven by induction on the length of $P''$, then on the length of $w$, recalling that residuals can be computed inductively via $\varepsilon^{-1}L = L$ and $(wb)^{-1}L = b^{-1}(w^{-1}L)$. When considering suffixes of $P'$ (or $\varnothing$), the following equalities can be used:

$$b^{-1}\varepsilon = \varnothing, \qquad\qquad b^{-1}\varnothing = \varnothing,$$

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

Note that the correctness of the third equality when $b = a$, and of the fourth equality when $b \in B$, rely on $b^{-1}P'' \subseteq P''$: this holds because $P'$, and then each suffix $P''$, is downward-closed.

Finally, $P'$ has at most $\ell + 1$ distinct non-empty residuals since it has $\ell + 1$ suffixes. Thus the minimal 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 [KP13] and conclude that $h(\downarrow P) \leqslant \ell + 1$.

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

The bounds in Proposition 5.2 can be reached, e.g., for $P = a \cdots a$.

**Corollary 5.3.** *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 $\downarrow_< L = \bigcup_i \downarrow_< P_i$. These unions are finite since there are only finitely many D-products of bounded length, so that we can invoke Eq. (2.6). □

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

**Theorem 5.4** (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* [BLS15]*) with $N$ nonterminals and $\ell = 4 \cdot 3^{N-1} + 2$.*

*Proof.* (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 $P_u \stackrel{\text{def}}{=} B_0^* a_1 B_1^* a_2 \ldots B_{p-1}^* a_p B_p^*$. 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 $\leqslant 2m + 1$. One now invokes Corollary 5.3.
(2) Bachmeier *et al.* showed that there is an NFA for $\downarrow L$ having $2 \cdot 3^{N-1}$ states [BLS15]. □

For our applications, we are interested in bounding $h(\downarrow L)$ in terms of $h(L)$ when $L$ is PT. Our main result in this section is:

**Theorem 5.5** (Downward closures of PT languages)**.** *Suppose that $L \subseteq A^*$ is PT and let $k = |A|$ and $m = f_k(h(L))$. Then*

$$h(\downarrow L) \leqslant (k + 1)(m + 1)\,, \qquad\qquad h(\downarrow_< L) \leqslant (k + 1)(m + 1)\,.$$

*Remark* 5.6. Before proving Theorem 5.5, let us observe that the upper bound it provides is quite good: for any $k, \lambda \geqslant 1$, the language $L = \{U_k\}$ from Section 3.1 has $h(U_k) = n = k\lambda + 1$ so that Theorem 5.5 gives $h(\downarrow U_k) < (k + 1)(\lambda + 2)^k$. On the other hand we know that $h(\downarrow U_k) = \lambda(\lambda + 1)^{k-1} + 1$ by Eq. (3.4).

The rest of this section is devoted to the proof of Theorem 5.5. Let $n = h(L)$ and $m = f_k(h(L))$. Our strategy is to cover $L$ by D-products of bounded length, relying on the fact that $L$ is closed under $\sim_n$.

**Lemma 5.7.** *For every $u \in A^*$ there is a D-product $P_u$ with at most $m$ letter factors and such that $u \in P_u \subseteq [u]_n$.*

*Proof.* By the small-subword theorem, $u$ has a subword $v = a_1 \cdots a_\ell$ with $v \sim_n u$ and $|v| = \ell \leqslant m$. Thus $u$ has the form

$$u \;=\; u_0\, a_1\, u_1\, a_2\, u_1 \cdots a_\ell\, u_\ell \;=\; b_{0,1} \cdots b_{0,p_0}\, a_1\, b_{1,1} \cdots b_{1,p_1}\, a_2 \cdots a_\ell\, b_{\ell,1} \cdots b_{\ell,p_\ell}\,. \tag{5.1}$$

Note that the above factorization is not necessarily unique, we just fix one. Then the $b_{i,j}$'s are the letters making up the $u_i$ factors in (5.1). To shorten notation, we let $a_0$ stand for $\varepsilon$ so that we can write $u = \prod_{i=0}^{\ell} a_i u_i = \prod_{i=0}^{\ell}(a_i \prod_{j=1}^{p_i} b_{i,j})$.

We claim that $P_u \stackrel{\text{def}}{=} \prod_{i=0}^{\ell}\bigl(a_i \prod_{j=1}^{p_i}\{b_{i,j}\}^*\bigr)$ proves the Lemma. That $u \in P_u$ and that $P_u$ has at most $m$ letter factors is clear. To show $P_u \subseteq [u]_n$, it is enough to invoke a natural generalisation of Lemma 2.1 (6) that, for the sake of completeness, we state and prove as Lemma 5.8. □

**Lemma 5.8.** *Assume $u = u_0u_1u_2\cdots u_\ell \lesssim_n u_0a_1u_1a_2u_2\cdots a_\ell u_\ell = v$ for some words $u_0,\ldots,u_\ell$ and letters $a_1,\ldots,a_\ell$. Then $u_0a_1^*u_1a_2^*u_2\cdots a_\ell^*u_\ell \subseteq [u]_n = [v]_n$.*

*Proof.* By induction on $\ell$. The case $\ell = 0$ is trivial so we assume $\ell > 0$. From $u \lesssim_n v$, and via Lemma 2.1 (1), we deduce $u' \lesssim_n v$ for $u' = u_0u_1a_2u_2\cdots a_\ell u_\ell$. With $u' \lesssim_n v$, Lemma 2.1 (6) gives

$$u_0a_1^*u_1a_2u_2\cdots a_\ell u_\ell \subseteq [u]_n = [u']_n = [v]_n\,. \tag{5.2}$$

Pick $k \in \mathbb{N}$ and write $w_k$ for $u_0a_1^ku_1a_2u_2\cdots a_\ell u_\ell$. Eq. (5.2) entails $w_k \sim_n u$. With $u \sqsubseteq u_0a_1^ku_1u_2\cdots u_\ell \sqsubseteq w_k$, and since $u \sim_n v \sim_n w_k$, Lemma 2.1 (1) now gives $u_0a_1^ku_1u_2\cdots u_\ell \lesssim_n w_k$. With the induction hypothesis, we obtain

$$u_0a_1^ku_1a_2^*u_2\cdots a_\ell^*u_\ell \subseteq [w_k]_n = [u]_n = [u_n]\,. \tag{5.3}$$

Since this holds for any $k \in \mathbb{N}$, we have proved the Lemma. □

After bounding the letter factors in the $P_u$'s (Lemma 5.7), we consider their star factors. For this it is convenient to write a D-product with $\ell$ letter factors under the form $P = \prod_{i=0}^{\ell}\bigl(a_i \prod_{j=1}^{p_i} B_{i,j}^*\bigr)$, i.e., regrouping the star factors in blocks separated by the letter factors, and again with $a_0$ standing for $\varepsilon$. With such a D-product, we associate $P' = \prod_{i=0}^{\ell} a_i\bigl(\prod_{j=1}^{p_i} B_{i,j}'^*\bigr)$ with the $B'_{i,j}$'s given by

$$B'_{i,j} \stackrel{\text{def}}{=} \bigl(B_{i,1} \cup B_{i,2} \cup \cdots \cup B_{i,j}\bigr) \cap \bigl(B_{i,j} \cup B_{i,j+1} \cup \cdots \cup B_{i,p_i}\bigr)\,. \tag{5.4}$$

That is, any star product $B_{i,j}$ is enlarged with letters that occur both on its left (inside the block of star products) and on its right. For example, with

$$P_0 = d^*\, a_1\, b^*\, (c + c')^*\, d^*\, (b + e)^*\, c^*\, a_2\, e^*\,,$$

we associate

$$P_0' = d^*\, a_1\, b^*\, (b + c + c')^*\, (b + c + d)^*\, (b + e + c)^*\, c^*\, a_2\, e^* \tag{†}$$

$$\equiv d^*\, a_1\, (b + c + c')^*\, (b + c + d)^*\, (b + e + c)^*\, a_2\, e^*\,. \tag{‡}$$

Since $B_{i,j} \subseteq B'_{i,j}$ for all $i,j$, one has $P \subseteq P'$. However $P'$ only enlarges $P$ in the following safe way:

**Lemma 5.9.** *For any $u \in A^*$ and $n \in \mathbb{N}$, $P \subseteq [u]_n$ implies $P' \subseteq [u]_n$.*

*Proof.* A word in $P'$ may use letters from some $B'_{i,j}$'s that are not in the corresponding $B_{i,j}$ and, with any $w \in P'$, we associate $\#_w$, the smallest number of letters that must be removed from $w$ before the resulting subword belongs to $P$.

We now assume $P \subseteq [u]_n$ and prove $w \in [u]_n$ for all $w \in P'$ by induction on $\#_w$. The base case where $\#_w = 0$ is trivial because then $w \in P$. So assume that $\#_w > 0$, i.e., $w \notin P$. Then there must exist in $w$ an occurrence of some letter $b$, from some $B'_{i,j}$, that does not appear in the corresponding $B_{i,j}$. (Note that, by Eq. (5.4), there must exist some $r < j$ and some $s > j$ such that $b \in B_{i,r} \cap B_{i,s}$.) Accordingly, we factor $w$ under the form

$$w = w_1\, \underline{a_i}\, \beta_1\, \underline{b}\, \beta_2\, \underline{a_{i+1}}\, w_2 \;,$$

highlighting the selected $b$, the occurrences of $a_i$ and $a_{i+1}$ that surround it, and where $\beta_1$ and $\beta_2$ belong to $B'^*_{i,1} \cdots B'^*_{i,j}$ and $B'^*_{i,j} \cdots B'^*_{i,p_i}$, respectively. Let $w' = w_1\, a_i\, \beta_1\, \beta_2\, a_{i+1}\, w_2$, i.e., $w'$ is $w$ without the $b \in B'_{i,j}$ that we singled out, so that $\#_{w'} < \#_w$ and the induction hypothesis yields $w' \in [u]_n$. We now claim that $w \sim_n w'$. To prove this, and since $w' \sqsubseteq w$, it is enough to show that any subword $t \in A^{\leqslant n}$ of $w$ is also a subword of $w'$. So consider one such $t$. From $t \sqsubseteq w$ we extract a factorization $t = t_1\, t_2\, t_3$ such that

$$t_1 \sqsubseteq w_1\, a_i\, \beta_1 \;, \qquad t_2 \sqsubseteq b \;, \qquad t_3 \sqsubseteq \beta_2\, a_{i+1}\, w_2 \;. \tag{5.5}$$

Thus $t_1 t_3 \sqsubseteq w'$ and, since $w' \in [u]_n$ and $|t_1 t_3| \leqslant n$, $t_1 t_3 \sqsubseteq x$ for any $x \in [u]_n$. In particular, $t_1 t_3 \sqsubseteq a_1 a_2 \cdots a_\ell \in P$, which requires $t_1 \sqsubseteq a_1 a_2 \cdots a_i$ or $t_3 \sqsubseteq a_{i+1} a_{i+2} \cdots a_\ell$ (or both). Let us assume $t_1 \sqsubseteq a_1 a_2 \cdots a_i$, the other case being similar. Combining with Eq. (5.5), we obtain

$$t = t_1 t_2 t_3 \sqsubseteq (w'' \stackrel{\text{def}}{=})\, a_1 a_2 \cdots a_i \cdot b \cdot \beta_2 a_{i+1} w_2 \;.$$

Note that $w'' \in P'$ and that $\#_{w''} < \#_w$ since the $b$ that follows $a_i$ in $w''$ can be accounted for by $B^*_{i,r}$. Thus $w'' \in [u]_n$ by induction hypothesis, i.e. $w'' \sim_n w'$, from which we deduce $t \sqsubseteq w'$. □

We can now bound the number of star factors in the D-product $P'$ associated with $P$. For this, we first simplify $P' = \prod_{i=0}^{\ell}\big(a_i \prod_{j=1}^{p_i} B'^*_{i,j}\big)$ by removing any $B'^*_{i,j}$ star factor that is subsumed by its immediate neighbour, i.e., such that $B'_{i,j} \subseteq B'_{i,j-1}$ or $B'_{i,j} \subseteq B'_{i,j+1}$. This is exactly how we moved from (†) to (‡) in our earlier example, and it shortens $P'$ without changing the denoted language. Once no more simplifications are possible, we can bound the length of the resulting D-product with the following combinatorial observation:

**Lemma 5.10.** *Assume that $A_1, \ldots, A_p \subseteq A$ are $p$ subalphabets such that*
— *for all $1 \leqslant j < p$, $A_j \nsubseteq A_{j+1}$ and $A_{j+1} \nsubseteq A_j$ ;*
— *for all $b \in A$ and $1 \leqslant j < k < j' \leqslant p$, if $b \in A_j \cap A_{j'}$, then $b \in A_k$.*
*Then $p \leqslant |A|$.*

*Proof.* Note that by the first condition, each $A_j$ is nonempty. Extend the sequence by defining $A_0 = A_{p+1} = \varnothing$. For $0 \leqslant j \leqslant p$, define $\Delta_j = A_j \triangle A_{j+1}$, where $\triangle$ denotes symmetric difference. Now $\Delta_0$ and $\Delta_p$ have size at least 1, and by the first condition, every other $\Delta_j$ has size at least 2. Thus $\sum_{j=0}^{p} |\Delta_j| \geqslant 2p$. By the second condition, any $b \in A$ occurs in at most two $\Delta_j$'s, thus $\sum_{j=0}^{p} |\Delta_j| \leqslant 2|A|$. So we conclude $2p \leqslant 2|A|$. □

**Corollary 5.11.** *For every $u \in A^*$ there is a D-product $P'_u$ of length at most $km + m + k$, and such that $u \in P'_u \subseteq [u]_n$.*

*Proof.* $P'_u$ as constructed above (and after simplifications) has at most $m$ letter factors, that separate at most $m+1$ blocks of star factors. Each such block is based on subalphabets $B'_{i,1}, \ldots, B'_{i,p_i}$ that satisfy the assumptions of Lemma 5.10: condition 1 holds since otherwise more simplifications could be performed, while condition 2 is a consequence of the definition of the $B'_{i,j}$'s via Eq. (5.4). Consequently, any star factor block in $P'_u$ has length at most $|A| = k$, leading to a $km+m+k$ bound for the total length of $P'_u$. Finally, that $u \in P'_u \subseteq [u]_n$ is a consequence of Lemma 5.9 since $u \in P_u \subseteq [u]_n$. □

We may now conclude the proof of Theorem 5.5. Indeed, with Corollary 5.11, and since $L$ is closed under $\sim_n$, we obtain $L = \bigcup_{u \in L} P'_u$, where each $P'_u$ has length bounded by $km + k + m$. We then apply Corollary 5.3.

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

Recall that $I(L)$ is the set of words which are incomparable (via $\sqsubseteq$ or $\sqsupseteq$) with *some word* in $L$. I.e., it is the image of $L$ by the incomparability relation $\perp$, or equivalently its pre-image since $\perp$ is symmetric.

In this section we prove the following result.

**Theorem 6.1.** *Suppose $L \subseteq A^*$ is PT and let $k = |A|$ and $m = f_k(h(L))$. Then $I(L)$ is PT and*

$$h(I(L)) \leqslant m + 1 \,.$$

We saw that $\perp_{A^*}$ is a rational relation, so that $I(L)$ is regular when $L$ is regular, see Proposition 2.1. Showing that $I$ also preserves piecewise-testability requires more work. For such questions, $I$ does not behave as simply as the pre-images we considered in earlier sections. In particular it does not necessarily yield languages that are PT, unlike $\uparrow L$ or $\downarrow L$.

At this point it is useful to examine some examples and make some general observations. Let $A = \{a, b, c\}$ and define the language $L_1$ of all finite prefixes of $(abc)^\omega$ via

$$L_1 = (abc)^*(\varepsilon + a + ab) = \{\varepsilon, a, ab, abc, abca, abcab, \ldots\} \,.$$

Note that $L_1$ is totally ordered by $\sqsubseteq$ hence no word of $L_1$ is in $I(L_1)$, i.e., $I(L_1) \subseteq A^* \smallsetminus L_1$.

To prove the reverse inclusion, we rely on the fact that a word is incomparable with any other word having same length. I.e., $I(u) \supseteq A^{=|u|} \smallsetminus \{u\}$ for any $u$, and thus, for any language $L$,

$$I(L) = \bigcup_{u \in L} I(u) \supseteq \bigcup_{u \in L} \bigl(A^{=|u|} \smallsetminus \{u\}\bigr) \,. \tag{6.1}$$

Since $L_1$ above contains at least one word of any given length, Eq. (6.1) entails $I(L_1) \supseteq A^* \smallsetminus L_1$. Finally we have proved that $I(L_1) = A^* \smallsetminus L_1$. Thus $I(L_1)$ is not PT since $L_1$ is not.

A similar example shows that $I(L)$ is not necessarily regular when $L$ is not. For example, take $A = \{a, b\}$ and let

$$L_2 = \{a^\ell b^\ell (\varepsilon + b) \mid \ell \in \mathbb{N}\} = \{\varepsilon, b, ab, abb, aabb, a^2b^3, a^3b^3, \ldots\} \,.$$

Here too $L_2$ is totally ordered by $\sqsubseteq$ and contains one word of each length. Hence $I(L_2) = A^* \smallsetminus L_2$, which is not regular.

Let us now consider some PT languages. In the case of a singleton language $L = \{w\}$, we know from Eq. (4.1) that $h(I(w)) = |w|$ when $|A| \geqslant 2$, and $h(I(w)) = 0$ when $|A| < 2$.

This can be used to bound $h(I(F))$ for a finite language $F$, using $I(F) = \bigcup_{w \in F} I(w)$.

Consider now $L_3 = [aab]_2$, i.e., $L_3 = aaa^*b$. This language is infinite but it is totally ordered by $\sqsubseteq$ and has one word of each length $\ell \geqslant 3$. Hence $I(L_3) = A^* \setminus L_3 \setminus {\downarrow}aab$ and we can easily bound $h(I(L_3))$ using results from the previous sections.

Another infinite PT language is $L_4 = [aabb]_2$, i.e., $L_4 = aaa^*bbb^*$. A different strategy applies here: $L_4$ contains no words of length $\ell < 4$ and exactly $\ell - 3$ words of each length $\ell \geqslant 4$. We may invoke a consequence of Eq. (6.1): if a language $L$ contains at least two words having same length $\ell$ then $I(L)$ contains all words of length $\ell$. Applied to $L_4$, this entails $I(L_4) \supseteq A^{\geqslant 5}$, which is enough to conclude that $I(L_4)$ is $A^{\geqslant 5} \cup F$ for some finite $F \subseteq A^{\leqslant 4}$, entailing $h(I(L_4)) \leqslant 5$.

As the above examples suggest, it is useful to think of the "layers" $L \cap A^{=\ell} = \{w \in L : |w| = \ell\}$ of $L$, and classify them into *empty*, *singular*, or *populous* layers, depending on whether they contain 0, 1, or more words. Observe that if $L \cap A^{=\ell}$ is populous then $I(L) \cap A^{=\ell}$ equals $A^{=\ell}$.

It is also useful to decompose PT-languages into the equivalence classes that make them up. Therefore, in the rest of this section, we focus on some equivalence class $[w]_n \subseteq A^*$ where $n = h(L)$.

A first observation is that the populous layers of $[w]_n$ propagate upwards:

**Lemma 6.2.** *Let $p \in \mathbb{N}$. If $[w]_n \cap A^{=p}$ is populous, then $[w]_n \cap A^{=p+1}$ is populous too.*

*Proof.* Assume that $[w]_n$ contains two different words $u_1, u_2$ of length $p$. Then $p > 0$ and these words can be written under the form $u_1 = u_0av_1$ and $u_2 = u_0bv_2$ where $u_0$ is their longest common prefix and $a, b$ are two distinct letters occurring at the first position where $u_1$ and $u_2$ differ. Applying Lemma 2.1 (3) we deduce that $[w]_n$ contains either $u_0abv_2$ or $u_0bav_1$. Let us assume, w.l.o.g., that $u_0bav_1 \sim_n w$ since the other case is similar. We now claim that $u_0bbv_2 \sim_n w$. Since $w \sim_n u_2 \sqsubseteq u_0bbv_2$, it is enough to show that every subword $s$ of $u_0bbv_2$ of length at most $n$ is also a subword of $u_2$. So let us pick any such $s$ and factor it as $s = s_0s_bs_2$ with $s_0 \sqsubseteq u_0$, $s_b \sqsubseteq bb$, and $s_2 \sqsubseteq v_2$, and with furthermore $s_0$ chosen longest possible. If $s_b \leqslant b$ then $s \sqsubseteq u_0bv_2 = u_2$ and we are done, so we assume $s_b = bb$. Let $s' = s_0bs_2$ and note that $s' \sqsubseteq u_2$, hence $s' \sqsubseteq u_1$ since $u_1 \sim_n u_2$ and $|s'| < |s| \leqslant n$. Now $s' = s_0bs_2 \sqsubseteq u_0av_1 = u_1$ requires $bs_2 \sqsubseteq av_1$ since our choice of $s_0$ longest entails $s_0b \not\sqsubseteq u_0$. This gives $s = s_0bbs_2 \sqsubseteq u_0bav_1$, hence $s \sqsubseteq u_2$ since $u_0bav_1 \sim_n u_2$.

We have shown that $[w]_n$ contains $u_0bav_1$ and $u_0bbv_2$, both words having length $p+1$. ☐

Populous layers also propagate downwards in the following sense:

**Lemma 6.3.** *Let $p \geqslant 2$. If $[w]_n \cap A^{=p}$ is populous then $[w]_n \cap A^{=p-1}$ is populous or $[w]_n \cap A^{=p-2}$ is empty (or both).*

*Proof.* Assume that layer $p$ is populous and that layer $p-2$ is non empty. If layer $p-2$ is populous, then layer $p-1$ is populous by Lemma 6.2 and we are done. So assume that $[w]_n \cap A^{=p-2} = \{x\}$ is singular. Then Lemma 2.1 (5) entails that $x \sqsubseteq w'$ for all $w' \in [w]_n \cap A^{\geqslant p-1}$, and since layer $p$ is not empty, Lemma 2.1 (1) entails that layer $p-1$ too is not empty. Pick $y \in [w]_n \cap A^{=p-1}$ and factor it as $y = uau'$ such that $x = uu'$. This entails $z = uaau' \in [w]_n \cap A^{=p}$ by Lemma 2.1 (6). By assumption, $[w]_n \cap A^{=p}$ is populous, hence contains another word $z' \neq z$ and there is a word $y' \in [w]_n \cap A^{=p-1}$ with $x \sqsubseteq y' \sqsubseteq z'$. If $y' \neq y$ we have proved that layer $p-1$ is populous and we are done. Otherwise $y \sqsubseteq z'$ and we may consider the different possibilities for $y = uau' \sqsubseteq z' \neq z$, knowing that $|z'| = |y| + 1$. If $z'$ is either some $uabu'$ or $ubau'$ with $b \neq a$ then we let $y'' = ubu'$. From $x = uu' \sqsubseteq y'' \sqsubseteq z'$ we deduce $y'' \in [w]_n$ by Lemma 2.1 (1). If $z'$ is $vau'$ with $u \sqsubset v$ we let $y'' = vu'$ and again deduce $y'' \in [w]_n$ from $x \sqsubseteq y'' \sqsubseteq z'$. (Note that $y'' \neq y$ since $vau' = z' \neq z = uaau'$.) If $z'$ is $uav'$ with $u' \sqsubset v'$, the same reasoning applies to $y'' = uv'$. In all three cases $y'' \neq y$ and $|y''| = |y| = p-1$, showing that $[w]_n \cap A^{=p-1}$ is populous. □

With Lemmas 6.2 and 6.3, we see that almost all layers of $[w]_n$ are populous as soon as one is. More precisely, when one layer is populous, either all (nonempty) layers are populous, or all are except for the lowest (nonempty) one. We already saw an example of the second situation with $L_4 = [aabb]_2 = aaa^*bbb^*$, and an example of the first situation is $L_5 = [abcab]_2$, where the lowest nonempty layer is $L_5 \cap A^{=5} = \{abcab, abcba, bacab, bacba\}$.

We now consider the general case:

**Lemma 6.4.** *$h(I([w]_n)) \leqslant m + 1$.*

*Proof.* Recall that $[w]_n$ is a singleton or is infinite (Lemma 2.1 (7)). We consider two cases.

(1) Assume that $[w]_n = \{w\}$ is a singleton. Then $h(I(w)) \leqslant |w|$ by Eq. (4.1) and $|w| \leqslant m$ by the small-subword theorem. Hence $h(I([w]_n)) \leqslant m$.

(2) Assume that $[w]_n$ is infinite. Let $u$ be a shortest word in $[w]_n$ and write $p$ for its length $|u|$. By the small-subword theorem, $p \leqslant m$. Since $[w]_n$ is infinite, and by Lemma 2.1 (1), it contains at least one word of each length $\geqslant p$, hence $I([w]_n) \supseteq A^* \smallsetminus [w]_n \smallsetminus {\downarrow_<} u$ by Eq. (6.1). There are two subcases.
   - (a) If $[w]_n$ is a total order under $\sqsubseteq$, no layer is populous, hence $I([w]_n) = A^* \smallsetminus [w]_n \smallsetminus {\downarrow_<} u$. Since $h({\downarrow_<} u) = |u| \leqslant m$ and $h([w]_n) = n \geqslant m$, we obtain $h(I([w_n])) \leqslant m$.
   - (b) If $[w]_n$ is not a total order under $\sqsubseteq$, all layers above $p$ are populous by Lemmas 6.2 and 6.3. If $u$ is the unique shortest word in $[w]_n$, we have $I([w]_n) = A^* \smallsetminus {\downarrow} u$, entailing $h(I([w]_n)) \leqslant |u| + 1 \leqslant m + 1$. Otherwise layer $p$ is populous too, entailing $A^{\geqslant p} \subseteq I([w]_n)$, i.e., $I([w]_n) = A^{\geqslant p} \cup F$ for some finite $F \subseteq A^{<p}$. We deduce $h(I([w]_n)) \leqslant p \leqslant m$ by Eq. (3.3). □

We may now conclude:

*Proof of Theorem 6.1.* Being $n$-PT, $L$ is a finite union $[w_1]_n \cup \cdots \cup [w_\ell]_n$ of equivalence classes of $\sim_n$, so that $I(L) = I([w_1]_n) \cup \cdots \cup I([w_\ell]_n)$. Now each $I([w_i]_n)$ is $(m+1)$-PT by Lemma 6.4 so that $I(L)$ is too. □

*Remark* 6.5. The upper bound in Theorem 6.1 is quite good: for any $k, \lambda \geqslant 1$, the language $L = \{U_k\}$ from Section 3.1 has $h(U_k) = n = k\lambda + 1$ so that Theorem 6.1 gives $h(I(U_k)) \leqslant (\lambda+2)^k$. On the other hand we know by Eq. (4.1) that $h(I(U_k)) = |U_k| = (\lambda+1)^k - 1$ when $k > 1$.

## 7. Deciding the two-variable logic of subwords

In this section we use our results on PT heights to establish complexity bounds on a decidable fragment of $\mathsf{FO}(A^*, \sqsubseteq)$, the first-order logic of subwords.

We assume familiarity with basic notions of first-order logic as exposed in, e.g., [Har09]: 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. In particular, if $\phi(x_1, \ldots, x_n)$ has $n$ free variables, we write $R_\phi$ for the $n$-ary relation defined by $\phi$ on the underlying structure.

The signature of the $\mathsf{FO}(A^*, \sqsubseteq)$ logic only contains one predicate symbol, "$\sqsubseteq$", denoting the subword relation. Terms are variables taken from a countable set $X = \{x, y, z, \ldots\}$ and all words $w_1, w_2, \ldots \in A^*$ 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 $x \mapsto bcab$.

The logic of the subword relation is a logic of substructure ordering like those considered by Ježek and McKenzie (see [JM09] and subsequent papers). It is one of the simplest and most natural substructure ordering occurring in computer science [Kus06]. In its full generality, this logic is computably isomorphic with $\mathsf{FO}(\mathbb{N}, +, \times)$, hence undecidable [KSY10]. We showed that already the $\Sigma_2$ fragment is undecidable [KS15] and recently Halfon *et al.* showed that even the $\Sigma_1$ fragment is undecidable [HSZ17]. This was very surprising: by comparison, "words equations", i.e., the $\Sigma_1$ fragment of $\mathsf{FO}(A^*, \cdot, =)$ in which the prefix relation can be defined, are decidable in $\mathsf{PSPACE}$ [Die02, Pla04, Jeż16].

We have previously shown that $\mathsf{FO}^2(A^*, \sqsubseteq)$, the 2-variable fragment, is decidable by a quantifier elimination technique [KS15]. In this article we extend our earlier analysis of the expressive power and complexity of the $\mathsf{FO}^2$.

When performing quantifier elimination, it is convenient to enrich the basic logic by allowing all regular languages $L_1, L_2, \ldots \in \mathsf{Reg}(A^*)$ as monadic predicates with the expected semantics, and we shall temporarily adopt this extension. We write $x \in L$ rather than $L(x)$ and assume that $L$ is given via a regular expression or a finite automaton — For example, we can state that $(a+b)^*$ is the downward closure of $(ab)^*$ with $\forall x\big[x \in (a+b)^* \iff \exists y(y \in (ab)^* \wedge x \sqsubseteq y)\big]$.

7.1. **Subword-recognizable relations.** In order to characterise the $\mathsf{FO}^2$-definable relations, we need some definitions. A relation $R \subseteq A^* \times A^*$ is *subword-recognizable*, if it belongs to the boolean closure of $\mathsf{Rec}(A^* \times A^*) \cup \{\sqsubseteq_{A^*}, \sqsupseteq_{A^*}\}$. It is furthermore *subword-piecewise-testable*, if it belongs to the boolean closure of $\mathsf{PT}(A^* \times A^*) \cup \{\sqsubseteq_{A^*}, \sqsupseteq_{A^*}\}$. We write $\mathsf{Rec}_{\sqsubseteq}(A^* \times A^*)$ and $\mathsf{PT}_{\sqsubseteq}(A^* \times A^*)$ for the corresponding classes.

**Proposition 7.1** (Normal form for $\mathsf{Rec}_{\sqsubseteq}(A^* \times A^*)$ and $\mathsf{PT}_{\sqsubseteq}(A^* \times A^*)$)**.** *A relation $R \subseteq A^* \times A^*$ is subword-recognizable if, and only if, it can be written under the form*

$$R \;=\; (\sqsubset_{A^*} \cap R_1) \cup (=_{A^*} \cap R_2) \cup (\sqsupset_{A^*} \cap R_3) \cup (\perp_{A^*} \cap R_4) \tag{NF}$$

*for some recognizable relations $R_1, R_2, R_3, R_4$.*

*Furthermore, $R$ is subword-piecewise-testable if, and only if, the relations $R_1, R_2, R_3, R_4$ can be chosen among the piecewise-testable relations.*

*Proof.* That the normal forms are subword-recognizable is clear since $=_{A^*}$ and $\sqsubset_{A^*}$ belong to the boolean closure of $\{\sqsubseteq_{A^*}, \sqsupseteq_{A^*}\}$: they are $\sqsubseteq_{A^*} \cap \sqsupseteq_{A^*}$ and $\sqsubseteq_{A^*} \smallsetminus =_{A^*}$.

Showing the other direction, i.e., that a subword-recognizable or subword-piecewise-testable relation $R$ can be put in normal forms, is done by induction on the boolean combination realising $R$ from the generators of the boolean closure. Here it suffices to show that normal forms are closed under intersections and complementations. Let us write $\{\xi_1, \xi_2, \xi_3, \xi_4\}$ for $\{\sqsubset_{A*}, =_{A*}, \sqsupset_{A*}, \perp_{A*}\}$ and $\bigcup_{i=1}^4 \xi_i \cap R_i$ for normal forms. Since the $\xi_i$'s are pairwise disjoint, we have

$$\Big(\bigcup_{i=1}^{4} \xi_i \cap R_i\Big) \cap \Big(\bigcup_{i=1}^{4} \xi_i \cap R'_i\Big) \;=\; \bigcup_{i=1}^{4} \xi_i \cap (R_i \cap R'_i)\,. \tag{7.1}$$

Since the $\xi_i$'s form a partition of $A^* \times A^*$, we further have

$$(A^* \times A^*) \smallsetminus (\xi_i \cap R) \;=\; \Big(\xi_i \cap \big[(A^* \times A^*) \smallsetminus R\big]\Big) \cup \bigcup_{j \neq i} \xi_j \cap (A^* \times A^*)\,. \tag{7.2}$$

Thus we see that normal forms are closed under boolean operations since $\mathsf{Rec}(A^* \times A^*)$ is.

Finally, the same proof applies to $\mathsf{PT}_{\sqsubseteq}(A^* \times A^*)$. □

7.2. **Quantifier elimination for $\mathsf{FO}^2(A^*, \sqsubseteq, L_1, L_2, \ldots)$.** We may now characterise the $\mathsf{FO}^2$-definable relations.

**Theorem 7.2.** *(i) A relation $R \subseteq A^* \times A^*$ is definable in the extended logic $\mathsf{FO}^2(A^*, \sqsubseteq, L_1, L_2, \ldots)$ iff it is subword-recognizable.*

*(ii) It is definable in the basic logic $\mathsf{FO}^2(A^*, \sqsubseteq, w_1, w_2, \ldots)$ iff it is subword-piecewise-testable.*

*(iii) Furthermore, a normal form for $R_\phi$ can be computed from the $\mathsf{FO}^2$ formula $\phi(x, y)$,*

*Proof.* The ($\Leftarrow$) direction of *(i)* is obvious: for example $R = \sqsubset_{A*} \cap L \times L'$ is definable via $x \sqsubseteq y \wedge x \neq y \wedge x \in L \wedge y \in L'$, a $\mathsf{FO}^2$ formula. When proving the same ($\Leftarrow$) direction for *(ii)*, we cannot use regular predicates to express $x \in L$ or $y \in L$. But since we assume that $L$ is PT, it is a boolean combination of filters $\uparrow u$, $\uparrow u'$, etc., so $x \in L$ can be expressed as a boolean combination of atomic formulas $u \sqsubseteq x$, $u' \sqsubseteq x$, etc.

We prove the ($\Rightarrow$) direction for *(i-ii)* by structural induction on the $\mathsf{FO}^2$-formula $\phi(x, y)$ that defines $R_\phi$. We consider all cases:

If $\phi$ is an atomic formula of the form $x \in L$ or $y \in L'$, then $R_\phi$ is $L \times A^*$ or $A^* \times L'$. If $\phi$ is some $u \in L$, then $R_\phi$ is one of the trivial $\varnothing$ or $A^* \times A^*$.

If $\phi$ is an atomic formula of the form $x \sqsubseteq u$ or $v \sqsubseteq y$ for some constant word $u$ or $v$ then $R_\phi$ is $(\downarrow u) \times A^*$ or $A^* \times (\uparrow v)$ respectively (a piecewise-testable relation in each case). If $\phi$ is $x \sqsubseteq y$ then $R_\phi$ is $\sqsubseteq_{A^*}$, and $R_\phi$ is trivial if $\phi$ is some $u \sqsubseteq v$.

If $\phi$ is a conjunction $\phi_1 \wedge \phi_2$ or a negation $\neg\phi_1$, we rely on the induction hypothesis and the closure properties of $\mathsf{Rec}_{\sqsubseteq}(A^* \times A^*)$ and $\mathsf{PT}_{\sqsubseteq}(A^* \times A^*)$.

The remaining case is when $\phi(x, y)$ is some $\exists x.\psi(x, y)$, the case $\exists y.\psi(x, y)$ being identical. Here $R_\phi = A^* \times \pi_2(R_\psi)$ where $\pi_2 : A^* \times A^* \to A^*$ is the projection $(u, v) \mapsto v$ lifted from pairs to sets of pairs, i.e., relations. To compute $\pi_2(R_\psi)$ we write $R_\psi$ in normal form —thanks to the induction hypothesis— and use $\pi_2\big(\bigcup_{i=1}^4 \xi_i \cap R_i\big) = \bigcup_{i=1}^4 \pi_2\big(\xi_i \cap R_i\big)$ and the following equalities (proofs omitted):

$$\pi_2(=_{A*} \cap L \times L') = L \cap L'\,, \qquad \pi_2(\sqsubset_{A*} \cap L \times L') = (\uparrow_{<} L) \cap L'\,, \tag{7.3}$$

$$\pi_2(\perp_{A*} \cap L \times L') = I(L) \cap L'\,, \qquad \pi_2(\sqsupset_{A*} \cap L \times L') = (\downarrow_{<} L) \cap L'\,. \tag{7.4}$$

Finally, in the case where $\phi$ does not use regular predicates, the above inductive construction only produces subword-piecewise-testable relations.

We now see why *(iii)* holds: the operations used above can all computed effectively on relations in normal form, e.g., using a quadruplet of automata for the $R_1, R_2, R_3, R_4$ of (NF). The required operations on automata are classic constructions: boolean combinations and images of regular languages by $\sqsupset_A$, $\sqsubset_A$ and $\perp_A$ (all rational relations). □

We note that if $\phi(x)$ is a $\mathsf{FO}^2$ formula with a single free variable, $R_\phi$ can be put under the form $L \times A^*$, i.e., $\phi(x)$ defines a regular property of words, and a piecewise-testable one if $\phi(x)$ is in the basic logic.

Theorem 7.2 has several corollaries. Firstly, and since the normal forms can be effectively computed, we have

**Corollary 7.3** (Decidability [KS15])**.** *Validity and satisfiability are decidable for* $\mathsf{FO}^2(A^*, \sqsubseteq, L_1, L_2, \ldots)$.

By contrast, note that the $\mathsf{FO}^3 \cap \Sigma_2$ and the $\Sigma_1$ fragments of the basic logic are undecidable [KS15, HSZ17].

Secondly, the computations inside the proof of Theorem 7.2 can be seen as a quantifier-elimination procedure.

**Corollary 7.4.** *Any* $\mathsf{FO}^2(A^*, \sqsubseteq, L_1, L_2, \ldots)$ *formula is effectively equivalent to a quantifier-free* $\mathsf{FO}^2$ *formula. The same holds for the basic logic* $\mathsf{FO}^2(A^*, \sqsubseteq, w_1, w_2, \ldots)$.

7.3. **Complexity for $\mathsf{FO}^2(A^*, \sqsubseteq)$.** The algorithm underlying the proof of Theorem 7.2 *(iii)* can be implemented using finite-state automata to handle and compute subword-recognizable relations via their normal forms. The steps described in Eqs. (7.3–7.4) involve computing images of regular languages by (fixed) rational relations and may induce an exponential complexity blow-up. In particular the pre-images $\uparrow_< L$ and $\downarrow_< L$ can have exponential size if one uses deterministic or alternating automata, while if one uses nondeterministic (or unambiguous) automata, the dual pre-images $\neg(\uparrow_<(\neg L))$ and $\neg(\downarrow_<(\neg L))$ —used in eliminating a universal quantifier— can have doubly exponential size [KNS16]. Therefore the best known upper bound for the decidability of $\mathsf{FO}^2(A^*, \sqsubseteq, L_1, L_2, \ldots)$ is a tower of exponentials with height bounded by the nesting depth of the formula at hand, hence a nonelementary complexity. Regarding lower bounds, only $\mathsf{PSPACE}$-hardness has been established [KS15].

We now turn to the basic logic, $\mathsf{FO}^2(A^*, \sqsubseteq, w_1, w_2, \ldots)$ where regular predicates are not allowed. As stated in Theorem 7.2, the quantifier-elimination procedure will only produce subword-piecewise-testable relations and languages. Furthermore, it is possible to bound the PT height of the defined languages and deduce an elementary complexity upper bound.

**Theorem 7.5** ($\mathsf{FO}^2(A^*, \sqsubseteq, w_1, w_2, \ldots)$ has elementary complexity)**.** *Assume that* $\phi$ *is a* $\mathsf{FO}^2(A^*, \sqsubseteq, w_1, w_2, \ldots)$ *formula. Then* $h(R_\phi)$ *is in* $2^{2^{O(|\phi|)}}$.
*Furthermore, computing DFAs for the normal form of* $R_\phi$ *(hence deciding the satisfiability or the validity of* $\phi$*) can be done in* $3{-}\mathsf{EXPTIME}$.

*Proof.* The quantifier-elimination procedure that proves Theorem 7.2 *(ii)* builds, for any subformula $\psi$ of $\phi$, a relation of the form $\bigcup_{i=1}^4 \xi_i \cap R_i$ represented by a quadruplet $R_1, R_2, R_3, R_4$ of PT relations. The PT height of these relations can be bounded. For example, the PT height is given by $|u|$ for $\psi$ an atomic formula of the form $u \sqsubseteq x$. Boolean combinations do not increase PT height even in the case of subword-piecewise-testable relations, see Eqs. (7.1) and (7.2). Quantifier-elimination can increase PT height when we compute $\uparrow_< L$, $\downarrow_< L$ and $I(L)$ as prescribed by Eqs. (7.3) and (7.4). But Theorems 4.4, 5.5 and 6.1 apply and show that the increase is polynomially bounded. Such increases combined at most $|\phi|$ times give a PT height bounded in $2^{2^{O(|\phi|)}}$.

Finally, when the PT height of $R_\phi$ (and of all intermediary $R_\psi$) have been bounded in $2^{2^{O(|\phi|)}}$, we obtain a bound on the size of the minimal DFAs and the time and space needed to compute them using Theorem 2.3. □

## 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 regular and context-free languages but more classes can be considered [Zet15]. 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, 5.5, and 6.1, and enriching the known collection of PT preserving operations.

## Acknowledgments

We thank the anonymous reviewers for their many helpful suggestions.

## Appendix A. Bounding $f_k(n)$

From Section 3, recall the definition of $f_k : \mathbb{N} \to \mathbb{N}$ where $k$ is a strictly positive integer:

$$f_1(n) = n\,, \tag{3.10}$$

$$f_{k+1}(n) = \max_{0 \leqslant m \leqslant n} m f_k(n+1-m) + m + f_k(n-m)\,. \tag{3.11}$$

We note that each $f_k$ function is monotonic: this is clear for $f_1$, and Eq. (3.11) guarantees that $f_{k+1}$ is monotonic if $f_k$ is.

In this appendix we prove the bound on $f_k(n)$ claimed in Section 3:

$$f_k(n) \leqslant \Big(\frac{n+2k-1}{k}\Big)^k - 1 < \Big(\frac{n}{k}+2\Big)^k\,. \tag{3.12}$$

To prove Eq. (3.12) we introduce the following auxiliary functions, where $0 < k \in \mathbb{N}$ and $x, y \in \mathbb{R}$:

$$F_k(x) \stackrel{\text{def}}{=} \left(\frac{x + 2k - 1}{k}\right)^k , \tag{$F$}$$

$$G_{k,x}(y) \stackrel{\text{def}}{=} (y+1)F_k(x - y + 1) = \frac{(y+1)(x - y + 2k)^k}{k^k} . \tag{$G$}$$

Let us check that $G_{k,x}\left(\frac{k+x}{k+1}\right) = F_{k+1}(x)$ for any $k > 0$ and $x \geqslant 0$:

$$\begin{aligned} G_{k,x}\left(\frac{k+x}{k+1}\right) &= \left(\frac{k+x}{k+1} + 1\right)\frac{1}{k^k}\left(x - \frac{k+x}{k+1} + 2k\right)^k \\ &= \frac{x+2k+1}{k+1}\ \frac{1}{k^k}\left(\frac{kx + 2k^2 + k}{k+1}\right)^k \\ &= \frac{x+2k+1}{k+1}\ \frac{1}{k^k}\left(\frac{k}{k+1}\right)^k (x + 2k + 1)^k \\ &= \left(\frac{x+2k+1}{k+1}\right)^{k+1} = F_{k+1}(x) . \end{aligned} \tag{$*$}$$

**Lemma A.1.** *$F_{k+1}(x) \geqslant G_{k,x}(y)$ for all $y \in [0, x]$.*

*Proof.* $G_{k,x}$ is well-defined and differentiable over $\mathbb{R}$, its derivative is

$$\begin{aligned} G'_{k,x}(y) &= \frac{(x - y + 2k)^k - (y+1)k(x - y + 2k)^{k-1}}{k^k} \\ &= \frac{(x - y + 2k)^{k-1}}{k^k}\big((x - y + 2k) - (y+1)k\big) \\ &= \frac{(x - y + 2k)^{k-1}}{k^k}\big(x + k - y(k+1)\big) . \end{aligned}$$

Thus $G'_{k,x}(y)$ is 0 for $y = y_{\max} \stackrel{\text{def}}{=} \frac{k+x}{k+1}$, is strictly positive for $0 \leqslant y < y_{\max}$, and strictly negative for $y_{\max} < y \leqslant x$. Hence, over $[0, x]$, $G_{k,x}$ reaches its maximum at $\frac{k+x}{k+1}$ and $(*)$ concludes the proof. ☐

**Proposition A.2.** *$f_k(n) + 1 \leqslant F_k(n)$ for all $k, n \in \mathbb{N}$ with $k > 0$.*

*Proof.* By induction on $k$. For the base case $k = 1$, one has $f_1(n) + 1 = n + 1 = F_1(n)$ by combining Eqs. (3.10) and $(F)$. For the inductive case $k \geqslant 2$, we know by Eq. (3.11) that

$$f_k(n) + 1 = m \cdot f_{k-1}(n + 1 - m) + m + f_{k-1}(n - m) + 1$$

for some $m \in \{0, \dots, n\}$,

$$\leqslant (m+1)\big[f_{k-1}(n + 1 - m) + 1\big]$$

by monotonicity of $f_{k-1}$,

$$\leqslant (m+1)F_{k-1}(n + 1 - m)$$

by induction hypothesis,

$$= G_{k-1,n}(m) \leqslant F_k(n)$$

by Eq. $(G)$ and Lemma A.1. ☐

This entails $f_k(n) \leqslant F_k(n) - 1$ which is exactly our original claim.

## References


[Ber79] J. Berstel. *Transductions and Context-Free Languages.* B. G. Teubner, Stuttgart, 1979.

[BLS15] 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.

[BSS12] M. Bojańczyk, L. Segoufin, and H. Straubing. Piecewise testable tree languages. *Logical Methods in Comp. Science*, 8(3), 2012.

[CP18] O. Carton and M. Pouzet. Simon's theorem for scattered words. In *Proc. DLT 2018*, volume 11088 of *Lecture Notes in Computer Science*, pages 182–193. Springer, 2018.

[DGK08] 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.

[Die02] V. Diekert. Makanin's algorithm. In M. Lothaire, editor, *Algebraic Combinatorics on Words*, volume 90 of *Encyclopedia of Mathematics and Its Applications*, chapter 13, pages 387–442. Cambridge Univ. Press, 2002.

[FK18] L. Fleischer and M. Kufleitner. Testing Simon's congruence. In *Proc. MFCS 2018*, volume 117 of *Leibniz International Proceedings in Informatics*, pages 62:1–62:13, 2018.

[GS16] J. Goubault-Larrecq and S. Schmitz. Deciding piecewise testable separability for regular tree languages. In *Proc. ICALP 2016*, volume 55 of *Leibniz International Proceedings in Informatics*, pages 97:1–97:15, 2016.

[Hai69] L. H. Haines. On free monoids partially ordered by embedding. *Journal of Combinatorial Theory*, 6(1):94–98, 1969.

[Har09] J. Harrison. *Handbook of Practical Logic and Automated Reasoning.* Cambridge University Press, 2009.

[Hig52] G. Higman. Ordering by divisibility in abstract algebras. *Proc. London Math. Soc. (3)*, 2(7):326–336, 1952.

[HM15] 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.

[HP00] K. Henckell and J.-E. Pin. Ordered monoids and $\mathcal{J}$-trivial monoids. In J.-C. Birget et al., editor, *Algorithmic Problems in Groups and Semigroups*, Trends in Mathematics, pages 121–136. Birkhäuser, 2000.

[HS19] S. Halfon and Ph. Schnoebelen. On shuffle products, acyclic automata and piecewise-testable languages. *Information Processing Letters*, 145:68–73, 2019.

[HSZ17] S. Halfon, Ph. Schnoebelen, and G. Zetzsche. Decidability, complexity, and expressiveness of first-order logic over the subword ordering. In *Proc. LICS 2017*, pages 1–12. IEEE, 2017.

[Jeż16] A. Jeż. Recompression: A simple and powerful technique for word equations. *Journal of the ACM*, 63(1):4:1–4:51, 2016.

[JM09] J. Ježek and R. McKenzie. Definability in substructure orderings, I: finite semilattices. *Algebra universalis*, 61(1):59, 2009.

[KCM08] L. Kontorovich, C. Cortes, and M. Mohri. Kernel methods for learning languages. *Theoretical Computer Science*, 405(3):223–236, 2008.

[KKS15] 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.

[Klí11] O. Klíma. Piecewise testable languages via combinatorics on words. *Discrete Mathematics*, 311(20):2124–2127, 2011.

[KNS16] 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.

[KP13] 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.

[KS15] 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.

[KS16] P. Karandikar and Ph. Schnoebelen. The height of piecewise-testable languages with applications in logical complexity. In *Proc. CSL 2016*, volume 62 of *Leibniz International Proceedings in Informatics*, pages 37:1–37:22, 2016.

[KSY10] O. V. Kudinov, V. L. Selivanov, and L. V. Yartseva. Definability in the subword order. In *Proc. CiE 2010*, volume 6158 of *Lecture Notes in Computer Science*, pages 246–255. Springer, 2010.

[Kus06] D. Kuske. Theories of orders on the set of words. *RAIRO Theoretical Informatics and Applications*, 40(1):53–74, 2006.

[Mas16] T. Masopust. Piecewise testable languages and nondeterministic automata. In *Proc. MFCS 2016*, volume 58 of *Leibniz International Proceedings in Informatics*, pages 67:1–67:14, 2016.

[Mat98] 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.

[MT15] 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.

[MT17] T. Masopust and M. Thomazo. On Boolean combinations forming piecewise testable languages. *Theoretical Computer Science*, 682:165–179, 2017.

[Pin86] J.-É. Pin. *Varieties of Formal Languages*. Plenum, New-York, 1986.

[Pla04] W. Plandowski. Satisfiability of word equations with constants is in PSPACE. *Journal of the ACM*, 51(3):483–496, 2004.

[PP04] D. Perrin and J.-É. Pin. *Infinite words: Automata, Semigroups, Logic and Games*, volume 141 of *Pure and Applied Mathematics Series*. Elsevier Science, 2004.

[PW97] J.-É. Pin and P. Weil. Polynomial closure and unambiguous product. *Theory of Computing Systems*, 30(4):383–422, 1997.

[RHF+13] J. Rogers, J. Heinz, M. Fero, J. Hurst, D. Lambert, and S. Wibel. Cognitive and sub-regular complexity. In *Proc. FG 2012 & 2013*, volume 8036 of *Lecture Notes in Computer Science*, pages 90–108. Springer, 2013.

[Sak09] J. Sakarovitch. *Elements of Automata Theory*. Cambridge University Press, 2009.

[Sim72] I. Simon. *Hierarchies of Event with Dot-Depth One*. PhD thesis, University of Waterloo, Dept. Applied Analysis and Computer Science, Waterloo, ON, Canada, 1972.

[Sim75] 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.

[Sim03] I. Simon. Words distinguished by their subwords. In *Proc. WORDS 2003*, 2003.

[SS83] 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.

[ST88] H. Straubing and D. Thérien. Partially ordered finite monoids and a theorem of I. Simon. *Journal of Algebra*, 119(2):393–399, 1988.

[SWY04] 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.

[Thé81] D. Thérien. Classification of finite monoids: The language approach. *Theoretical Computer Science*, 14(2):195–208, 1981.

[Tra01] A. N. Trahtman. Piecewise and local threshold testability of DFA. In *Proc. FCT 2001*, volume 2138 of *Lecture Notes in Computer Science*, pages 347–358. Springer, 2001.

[Yu05] Sheng Yu. State complexity: Recent results and open problems. *Fundamenta Informaticae*, 64(1–4):471–480, 2005.

[Zet15] 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.

[Zet18] G. Zetzsche. Separability by piecewise testable languages and downward closures beyond subwords. In *Proc. LICS 2018*, pages 929–938. ACM Press, 2018.