Hierarchy of piecewise-testable languages and complexity of the two-variable logic of subsequences

P. Karandikar\textsuperscript{1} and Ph. Schnoebelen\textsuperscript{2}

\textsuperscript{1} LIAFA, Université Paris Diderot, France
\textsuperscript{2} LSV, ENS Cachan, CNRS, Université Paris-Saclay, France

Abstract. We prove that $\text{FO}^2(A^*, \subseteq)$, the two-variable fragment of the first-order logic of sequences with the subsequence ordering, can only express piecewise-testable properties and is decidable with elementary complexity. To prove this we develop new techniques for bounding the piecewise-testability level of regular languages.

1 Introduction

A subsequence of a (finite) sequence $u = (a_1, \ldots, a_\ell)$ is a sequence obtained from $u$ by removing any number of elements. For example, if $u = (a, b, a, b, a)$ then $v = (b, b, a)$ is a subsequence of $u$, a fact we denote with $v \subseteq u$.

We use $\text{FO}(A^*, \subseteq)$ to denote the first-order logic where the universe consists of all finite sequences over a given set of atoms $A$ and where $\subseteq$ is the only predicate. A simple example of a $\text{FO}(A^*, \subseteq)$ sentence is $\exists x \forall y (x \subseteq y)$, stating the existence of a minimal sequence.

While the notion of subsequence is a fundamental one and occurs prominently in many parts of logic and computer science (in searching and pattern matching, coding theory, bioinformatics, …), not much is known about $\text{FO}(A^*, \subseteq)$. Kuske showed that the logic is undecidable as soon as $A$ contains two atoms, and that already its $\Sigma_3$ theory is undecidable [16]. The $\Sigma_1$ theory is decidable (in fact, $\text{NP}$-complete) and we showed recently that already the $\Sigma_2$ theory is undecidable, even when further restricting to the 3-variable fragment $\Sigma_2 \cap \text{FO}^3$ [11].

On the positive side, the 2-variable fragment $\text{FO}^2(A^*, \subseteq)$ is decidable. In fact, any $\text{FO}^2$ formula $\phi(x)$ with one free variable defines a regular set of sequences that can be computed effectively using automata-theoretic techniques [11]. Regarding complexity, only a $\text{PSPACE}$ lower bound has been proven while the automata-theoretic decision procedure has nonelementary complexity since every quantifier elimination may induce an exponential complexity blowup.

Our contribution. We characterise the sets of sequences (the languages) that can be defined in the $\text{FO}^2(A^*, \subseteq)$ fragment: they are exactly the piecewise-testable (PT) languages introduced by Simon [20], i.e., the finite boolean combinations of upward cones $\uparrow u$ (see definitions in Section 2). This result requires new techniques for showing that a language is PT, in particular for showing that if $L \subseteq A^*$
is PT then $I(L)$, the set of sequences that are incomparable with a sequence in $L$, is PT too, see Theorem 6.1.

We further use this characterisation to prove an elementary upper bound on the complexity of $\text{FO}^2(A^*, \sqsubseteq)$. Indeed, PT languages are arranged in a hierarchy according to their piecewise-testability level, or PT-level and this provides a very natural measure of descriptive complexity. Informally, the PT-level of $L$ is the minimal length of subwords needed to describe $L$. Little is known on how to measure PT-levels and prove upper and lower bounds on them. In this paper we introduce new combinatorial tools allowing to bound the PT-level of the languages constructed by the decision procedure for $\text{FO}^2(A^*, \sqsubseteq)$, resulting in the elementary $3\text{-EXPTIME}$ complexity stated in Theorem 3.3. Beyond this application to $\text{FO}^2(A^*, \sqsubseteq)$, the results and techniques we develop in this paper contribute to a larger research program whose goal is to use the PT hierarchy as an effective measure of descriptive complexity.

Related work. First-order logics of sequences usually do not include the subsequence predicate, they rather consider the prefix ordering, and/or functions for taking contiguous subsequences or computing the length of sequences, see, e.g., [4, 7]. In automated deduction and specifically in ordered constraints solving, the decidability of logics of simplification orderings on strings and trees — $\text{FO}(A^*, \sqsubseteq)$ being a special case— is a key issue [1]. These works often limit their scope to $\Sigma_1$ or similar fragments since decidability is elusive in this area. We must also mention that the structure $(A^*, \sqsubseteq)$ is a simple case of discrete structures well quasi-ordered by an embedding relation as they appear in the Graph Minor Theorem and many areas of graph theory.

Regarding PT languages, they have received significant attention since their introduction by Simon [20], see e.g., [3, 12, 13, 18, 19] and the references therein. They are connected with the Straubing-Thérien hierarchy, the dot-depth hierarchy, and have proved useful in database theory, machine learning, linguistics, etc. PT-levels are used to measure the difference between separable languages, see e.g. [2, 6] and references therein. The PT-level of a PT language is computable [17] but we know little on how this level relates to other features of the language. Klíma and Polák show that the PT-level of $L$ is bounded by the depth of the minimal DFA for $L$ [14]. They also show that for every $n$ there exists words $u$ of length $|u| = 4n^2$ and such that $\{u\}$ has PT-level $4n - 1$ [14]. In the other direction [8] counts the number of languages in a given level, implicitly providing bounds on the size of the canonical automaton for $L$ (see Theorem 2.2). Related bounds for the depth (not the size) of the automaton are given in [17].

2 Basic notions

Since our constructions heavily rely on concepts and results from formal language theory, we shall from now on speak of “words” and “letters” (from an “alphabet”) rather than sequences and atoms. For ease of exposition, we only consider finite alphabets like $A = \{a, b, c\}$: this is no real restriction since only finitely many letters appear in a given $\text{FO}(A^*, \sqsubseteq)$ formula.
We use $u, v, \ldots$ to denote words (finite sequences of letters) like $aac$. Concatenation is written multiplicatively, with the empty word $\varepsilon$ as unit. We freely use regular expressions like $(ab)^* + (ba)^*$ to denote regular languages.

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

A word $v$ is a factor of $u$ if there exist words $u_1$ and $u_2$ such that $u = u_1vu_2$. If furthermore $u_1 = \varepsilon$ then $v$ is a prefix of $u$ and we write $v^{-1}u$ to denote the residual $u_2$. There is a mirror notion of suffix.

Subwords. We say that a word $u$ is a subword (i.e., a subsequence) of $v$, written $u \subseteq v$, when $u$ is some $a_1 \cdots a_n$ and $v$ can be written as $v_0a_1v_1 \cdots a_nv_n$ for some $v_0,v_1,\ldots,v_n \in A^*$, e.g., $\varepsilon \subseteq bba \subseteq ababa$. We write $u \subseteq v$ 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 $u \nsubseteq v$ and $v \nsubseteq u$. Factors are a special case of subwords.

With a word $u \in A^*$ we associate the upward and downward cones, $\uparrow u$ and $\downarrow u$, defined with

$$\uparrow u = \{v \in A^* \mid u \subseteq v\}, \quad \downarrow u = \{v \in A^* \mid v \subseteq u\}.$$  

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

$$I(L) \defeq \{u \in A^* \mid \exists v \in L : u \perp v\}, \quad C(L) \defeq \{u \in A^* \mid L \subseteq \uparrow u \cup \downarrow u\}.$$  

Thus $I(L)$ collects all words that are incomparable with some word in $L$, and $C(L)$ coincides with $A^* \setminus I(L)$. Note that $C(L \cup L') = C(L) \cap C(L')$.

Simon’s congruence and piecewise-testable languages. For $n \geq 1$ and $u, v \in A^*$, we write $u \sim_n v$ if $u$ and $v$ have the same subwords of length up to $n$, i.e., if $\downarrow u \cap A^{\leq n} = \downarrow v \cap A^{\leq n}$. Note that $u \sim_n v$ and $u' \sim_n v'$ imply $uu' \sim_n vv'$, hence $\sim_n$ is called Simon’s congruence (for length $n$) \cite{19, 20}. 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.

A language $L \subseteq A^*$ is said to be $n$-piecewise testable ($n$-PT) if it is closed under $\sim_n$, i.e., if it is a finite union $\bigcup [u]_n$ of $\sim_n$ classes. It is piecewise testable

\footnote{The definition of $\uparrow u$ involves an implicit $A$ that will always be clear from the context.}
(PT) if it is $n$-PT for some $n$. Note that if $L$ is $n$-PT and $m > n$, then $L$ is also $m$-PT. We write $ptl(L)$ for the smallest $n$ —called the “PT-level” of $L$— such that $L$ is $n$-PT, letting $ptl(L) = \infty$ when $L$ is not PT.

It is convenient to introduce the notation $\uparrow u$ for $\{v \in A^* \mid u \subseteq v\}$, i.e. for $A^* \setminus \uparrow u$. While $\uparrow u$ collects all words having $u$ as a subword, $\uparrow u$ collects the words excluding $u$ as a subword. One may define any equivalence class $u$ with $u \in A^*$ with $u \not\sim v$.

One then sees that a language $L$ is $n$-PT if, and only if, it is a finite boolean combination $L = \bigcup (\bigcap_i \uparrow u_{i,j} \cap \bigcap_j \uparrow u_{i,j})$ where every $u_{i,j}$ and $v_{i,j}$ has length at most $n$. We see here that PT languages are a subclass of the regular languages, and indeed a subclass of the star-free languages.

**PT preserving operations.** It is well known that $n$-PT languages are closed under union, intersection, complements, and reversals. They are also closed under left or right residuals.

**Lemma 2.1 (Residuals).** If $L, M \subseteq A^*$ and $L$ is $n$-PT then $L / M$ is $n$-PT, where $L / M \eqdef \{u \mid \exists w \in M : wu \in L\}$.

**Proof.** Assume $u \in L / M$ and $u \sim_n v$. Then $wu \in L$ for some $w \in M$. Now $wu \sim_n wv$ thus $wv \in L$ since $L$ is closed under $\sim_n$, and then $v \in L / M$. We conclude that $L / M$ is closed under $\sim_n$. 

Thus the set of prefixes (also suffixes, or factors) of $L$ is $n$-PT when $L$ is.

Finally, two simple but important constructions that provide PT languages are the closures by subwords and superwords, $\uparrow L$ and $\downarrow L$, defined above. Every upward-closed language is PT since it is the union of finitely many languages of the form $\uparrow u$ (by Higman’s Lemma [15]). Every downward-closed language is PT too since its complement is upward-closed.

We are not aware of more piecewise-testability preserving operations on languages in the literature. On the negative side, let us recall that PT languages are not closed under concatenation —e.g., $a$ and $(a + b)^*$ are PT but $(a + b)^*a$ is not\(^4\) — or Kleene star —PT languages are star-free.—

**Relating PT-level and state complexity.** One can bound the PT-level of $L$ by the depth of its canonical DFA [14]. In the other direction, we can prove

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

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

\(^4\) It contains $(ba)^k$ but not $(ba)^k b$, however $(ba)^k \sim_k (ba)^k b$ for any $k$. 

In the above, log means $\log$ to the base 2. Thus, seeing $k$ as fixed, the canonical DFA for $L$ has $2^{O(n^{k-1}\log n)}$ states.

Proof. We build a DFA for $L$ which remembers the equivalence class under $\sim_n$ of the word it has read so far. This suffices because for all $w \in A^*$ and $a \in A$, the class $[wa]_n$ of $wa$ is determined by $[w]_n$ and $a$. The initial state is $[\varepsilon]_n$, and the final states are all the classes $[u]_n$ which are a subset of $L$. In [8] we showed that the number of equivalence classes of $\sim_n$ is bounded by $m$.

Remark 2.3. [17] shows that the depth (not the size) of the canonical DFA is bounded by $\binom{n+d}{n} - 1$.

3 Deciding the first-order logic of subwords

We assume familiarity with basic notions of first-order logic as exposed in, e.g., [5]: bound and free occurrences of variables, quantifier depth of formulae, and fragments $\text{FO}^n$ where at most $n$ different variables (free or bound) are used.

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

When deciding the $\text{FO}^2$ fragment, it is natural to enrich the basic logic by allowing all regular expressions as monadic predicates (with the expected semantics). For example, we can state that the downward closure of $(ab)^*$ is exactly $(a+b)^*$ with $\forall x \{x \in (a+b)^* \iff \exists y \in (ab)^* \land x \sqsubseteq y\}$.

When writing $\text{FO}(A^*, \sqsubseteq)$ formulae we freely use abbreviations like $x \sqsubseteq y$, $x \sqsupseteq y$, and $x \sqsubset y$ since they can be defined in terms of $\sqsubseteq$. Finally, we use negated predicate symbols as in $x \not\sqsubseteq y$ or $x \not\sqsubset (ab)^*$ with obvious meaning.

In [11] we showed that validity and satisfiability are decidable for the $\text{FO}^2$ fragment of the logic, even enriched with regular predicates (note that the $\text{FO}^2 \cap \Sigma_2$ fragment is undecidable). Since we later prove a complexity upper bound on the underlying algorithm, we need to recall the main lines of the decidability proof (see [11] for full details).

3.1 Decidability for $\text{FO}^2(A^*, \sqsubseteq)$

In the following we consider $\text{FO}^2$ formulae using only $x$ and $y$ as variables. We allow regular predicates of the form $x \in L$ and $y \in L'$ for given regular languages (i.e., we consider the enriched logic). Furthermore, we consider a variant of the logic where we use the binary relations $\sqsubseteq, \sqsupseteq, = \text{ and } \sqsubset$ instead of $\sqsubseteq$. This will be convenient later. The two variants are equivalent, even when restricting to $\text{FO}^m$ fragments, since the new set of predicates can be defined in terms of $\sqsubseteq$ and vice versa.
Lemma 3.1. Let $\phi(x)$ be an $\text{FO}^2(A^*, \sqsubseteq)$ formula with at most one free variable. Then there exists a regular language $L_\phi \subseteq A^*$ such that $\phi(x)$ is equivalent to $x \in L_\phi$. Furthermore, $L_\phi$ can be built effectively from $\phi$ and $A$.

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

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

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

$$\neg \psi \rightarrow \psi \quad \neg (\psi_1 \lor \psi_2) \rightarrow \neg \psi_1 \land \neg \psi_2 \quad \neg (\psi_1 \land \psi_2) \rightarrow \neg \psi_1 \lor \neg \psi_2$$

and then eliminating negations completely with:

$$\neg (z \in L) \rightarrow z \in (A^* \setminus L) \quad \neg (z_1 R_1 z_2) \rightarrow z_1 R_2 z_2 \lor z_1 R_3 z_2 \lor z_1 R_4 z_2$$

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

Thus, we may now assume that $\phi'$ is a positive boolean combination of atomic formulae. We write $\phi'$ in disjunctive normal form, that is, as a disjunction of conjunctions of atomic formulae. Observing that $\exists y (\phi_1 \lor \phi_2)$ is equivalent to $\exists y \phi_1 \lor \exists y \phi_2$, we assume w.l.o.g. that $\phi'$ is just a conjunction of atomic formulae.

Any atomic formula of the form $x \in L$, for some $L$, can be moved outside the existential quantification, since $\exists y (x \in L \land \psi)$ is equivalent to $x \in L \land \exists y \psi$. All atomic formulae of the form $y \in L$ can be combined into a single one, since regular languages are closed under intersection.

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

Theorem 3.2. \cite{11}. The truth problem for $\text{FO}^2(A^*, \sqsubseteq)$ is decidable.

\[\]

\[\]5 This is well known and easy to see for $\sqsubseteq$ and $\sqsupseteq$. It is proved in \cite{11} for $\perp$.\[\]
Proof. Lemma 3.1 provides a recursive procedure for computing the set of words that make \( \phi(x) \) true. When \( \phi \) is a closed formula, this set is \( A^* \) or \( \emptyset \) depending on whether \( \phi \) is true or not.

### 3.2 Complexity for \( \text{FO}^2(A^*, \sqsubseteq) \)

The algorithm underlying the proof of Lemma 3.1 can be implemented using finite-state automata to handle the regular languages \( L_\phi \) that are constructed for each subformula. However, complementation of nondeterministic automata is costly and one cannot circumvent this by using deterministic or alternating automata (DFAs or AFAs) since computing the pre-images \( \uparrow_L \) and \( \downarrow_L \) are costly for DFAs and AFAs, see [9] for details. Finally, the only clear upper bound for the algorithm is a tower of exponentials whose height is given by the quantifier depth of the formula at hand, hence a nonelementary complexity. Note that regarding lower bounds, [11] only proves \( \text{PSPACE} \)-hardness for the logic.

We now turn our attention to the basic logic \( \text{FO}^2(A^*, \sqsubseteq) \), where regular predicates are not allowed. It is easy to see that any piecewise-testable predicate can be defined in the logic, in fact as a quantifier-free formula with one free variable since for a PT language \( L \), “\( x \in L \)” is equivalent to a boolean combination of atomic formulae of the form \( w \sqsubseteq x \) for words \( w \) of length bounded by the PT-level of \( L \).

It turns out that even with two variables and first-order quantifiers, the logic can only express piecewise-testable predicates. Furthermore, it is possible to bound the PT-level of the defined languages, entailing an elementary complexity upper bound:

**Theorem 3.3 (Main Theorem).** If \( \phi(x) \) is an \( \text{FO}^2 \) formula in the basic logic, then \( L_\phi \) is a piecewise-testable language with a PT-level in \( 2^\mathcal{O}(|\phi|) \).

Furthermore, computing a canonical DFA for \( L_\phi \) (hence deciding the truth of \( \phi \)) can be done in \( 3\text{-EXPTIME} \).

Proof. We mimic the proof of Lemma 3.1. In this process we can allow atomic formulae “\( x \in L \)” when \( L \) is PT, since as observed earlier this can be expressed as a boolean combination of atomic formulae of the form \( w \sqsubseteq x \). The key extra ingredient is that the pre-images \( R^{-1}(L) \) preserve piecewise-testability and that \( \text{ptl}(R^{-1}(L)) \) is in \( O(\text{ptl}(L)^{|A|}) \): this is the subject of Theorem 4.5 for \( R = \sqsubseteq \), Theorem 5.1 for \( R = \sqsubseteq \), and Theorem 6.1 for \( R = \perp \).

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

### 4 Bounding piecewise-testability levels

We develop in this section our first tool for bounding the PT-level of the languages constructed in the proof of Theorem 3.3. In particular, Proposition 4.2
below shows that a class \([u]_n\) can always be described by a representative \(v\) of bounded size and that furthermore embeds in \(u\). This will be used repeatedly in the course of this paper.

Define \(u \preceq_n^1 v\) if all the following hold:
\[
\begin{align*}
&u \subseteq v, \\
&|u| = |v| - 1, \\
&u \sim_n v.
\end{align*}
\]
Let \(\preceq_n\) be the reflexive transitive closure of the relation \(\preceq_n^1\) on \(A^*\). Thus \(u \preceq_n v\) if \(u \subseteq v \sim_n u\) and there is a sequence \(u = u_0 \subseteq u_1 \subseteq \cdots \subseteq u_\ell = v\) such that all \(u_i\) are \(\sim_n\)-equivalent, and each \(u_{i+1}\) is obtained by inserting exactly one letter somewhere in \(u_i\). The following properties will be useful:

**Lemma 4.1.** For all \(u, v \in A^*\) and \(a \in A\):

1. If \(u \sim_n v\) then there exists \(w \in A^*\) such that \(u \preceq_n w\) and \(v \preceq_n w\).
2. If \(uv \sim_n uav\) then \(uv \sim_n uav\) for all \(\ell \in \mathbb{N}\).
3. Every equivalence class of \(\sim_n\) is a singleton or is infinite.

**Proof.** (1) is Lemma 6 from [20]; (2) is in the proof of [19, Coro. 2.8]; (3) follows from (1,2).

### 4.1 A generic upper bound

For \(k = 1, 2, \ldots\) define \(f_k : \mathbb{N} \rightarrow \mathbb{N}\) by induction on \(k\) with
\[
f_1(n) = n, \quad f_{k+1}(n) = \max_{m \leq n} m f_k(n + 1 - m) + m + f_k(n - m).
\]

The definition of \(f_k\) is only used in Appendix A. In the rest of the paper, we just use the following upper bound which is proved in [10, Prop. 4.4].

\[
f_k(n) < \left(\frac{n + 2k - 1}{k}\right)^k \leq \left(\frac{n}{2k + 2}\right)^k.
\]

**Proposition 4.2 (See Appendix A).** Let \(k = |A|\). For all \(u \in A^*\) and \(n \in \mathbb{N}\) there exists some \(v \in A^*\) with \(v \preceq_n u\) and such that \(|v| \leq f_k(n)\).

Note that the bound \(f_k(n)\) in Proposition 4.2 does not depend on \(u\).

We can already apply Proposition 4.2 to the case of finite languages. For a word of length \(|u| = \ell\), the singleton \(\{u\}\) can be defined as \(|u| \cap \bigcap_{v \in A^*} \omega e v,\). Hence \(ptl(\{u\}) \leq \ell + 1\).

**Proposition 4.3 (Finite languages).** Suppose \(L \subseteq A^*\) is finite and nonempty with \(|A| = k\). Let \(\ell\) be the length of the longest word in \(L\). Then \(1 + k(\ell^{1/k} - 1) < \text{ptl}(L) \leq \ell + 1\).

**Proof.** We may assume w.l.o.g. that \(L = \{u\}\) is a singleton. Let \(n = ptl(L)\), so that \(L = [u]_n\). By Proposition 4.2 and Eq. (2), \(\ell \leq f_k(n) < \left(\frac{n + 2k - 1}{k}\right)^k\). This gives \(n > 1 + k(\ell^{1/k} - 2)\).
We now claim that for any $k$,\( ptl\{a^i\} = \ell + 1 \). The lower bound is quite good: For arbitrarily large lengths $\ell$, we exhibit a singleton $L = \{u_k\}$ on a $k$-letter alphabet with $|u_k| = \ell$ such that $ptl(L) = 1 + k((\ell + 1)^{1/k} - 1)$.

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 number $\eta \in \mathbb{N}$. We let $u_0 \overset{\text{def}}{=} \varepsilon$ and, for $k > 0$, $u_k \overset{\text{def}}{=} (u_{k-1} a_k)^{\eta} u_{k-1}$. For example, for $\eta = 3$ and $k = 2$, one has $u_2 = a_1 a_1 a_2 a_1 a_1 a_2 a_1 a_1 a_1$.

**Proposition 4.4.** For $k \geq 0$, $|u_k| = (\eta + 1)^k - 1$ and $ptl\{u_k\} = k\eta + 1$.

**Proof sketch.** An easy induction on $k$ shows that $|u_k| = (\eta + 1)^k - 1$. That $ptl\{u_k\} = k\eta + 1$ is quite good: For arbitrarily large lengths $\ell$ one has $|u_k| \geq \ell - 1$ for all $\ell$. We define a word $u_k \in A_k^*$ by induction on $k$ and parameterized by a number $\eta \in \mathbb{N}$. We let $u_0 \overset{\text{def}}{=} \varepsilon$ and, for $k > 0$, $u_k \overset{\text{def}}{=} (u_{k-1} a_k)^{\eta} u_{k-1}$. For example, for $\eta = 3$ and $k = 2$, one has $u_2 = a_1 a_1 a_2 a_1 a_1 a_2 a_1 a_1 a_1$.

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

\[
\left( \bigwedge_{v \in P_k} v \sqsubseteq u \right) \land \left( \bigwedge_{w \in N_k} w \nmid u \right) \iff u = u_k. \tag{3}
\]

(See Appendix B for a proof.) Thus $ptl\{u_k\} \leq k\eta + 1$ since the words in $P_k$ have length $k\eta$ and the words in $N_k$ have length at most $k\eta + 1$.

It remains to show that $ptl\{u_k\} > k\eta$, i.e., that $\{u_k\}$ is not closed under $\sim_{k\eta}$; for this it is enough to note that $u_k \sim_{k\eta} u_k a_1$ using [20, Lemma 3].

For later use we also record the following bounds:

\[
\begin{align*}
\eta(\eta + 1)^{k-1} &\leq ptl(\uparrow u_k) \leq (\eta + 1)^k - 1, \tag{4} \\
\eta(\eta + 1)^{k-1} &< ptl(\downarrow u_k) \leq (\eta + 1)^k, \tag{5} \\
(\eta + 1)^{k-1} &< ptl(I(u_k)) \leq (\eta + 1)^k. \tag{6}
\end{align*}
\]

**Proof.** First note that $|u_k|_{a_1} = \eta(\eta + 1)^{k-1}$. Thus $a_1^\ell \sqsubseteq u_k$ iff $\ell \leq \eta(\eta + 1)^{k-1}$. Since $a_1^\ell \sim a_1^{\ell+1}$ for all $\ell$, we deduce the lower bounds for $ptl(\downarrow u_k)$ and $ptl(I(u_k))$.

Similarly, for a word $v = a_1 \cdots a_k$ using all letters once, $u_k \sqsubseteq v^\ell$ iff $\ell \geq \eta(\eta + 1)^{k-1}$. Since $v^\ell \sim v^{\ell+1}$ for all $\ell$, we deduce $ptl(\uparrow u_k) \geq \eta(\eta + 1)^{k-1}$.

The upper bounds come easily from $|u_k| = (\eta + 1)^k - 1$ since for any $u \in A^*$ one has $ptl(\uparrow u) \leq |u|$, $ptl(\downarrow u) \leq |u| + 1$ and $I(u) = A^* \setminus \uparrow u \setminus \downarrow u$. 

\[\square\]
4.2 Upward closures

It is known that $\uparrow L$ and $\downarrow L$ are PT for any $L$. Related languages are $\uparrow L$ and $\downarrow L$ (used in Theorem 3.3) as well as $\min(L) \overset{\text{def}}{=} \{ u \in L \mid \forall v \in L : v \subset u \}$.

**Theorem 4.5 (Upward closures).** Suppose $L \subseteq A^*$ is $n$-PT and $|A| = k$. Let $m = f_k(n)$. Then $\uparrow L$ is $m$-PT, while $\downarrow L$ and $\min(L)$ are $(m + 1)$-PT.

**Proof.** By Proposition 4.2, the minimal elements of $L$ have length bounded by $m$. This proves $\ptl(\uparrow L) \leq m$ and, using Proposition 4.3, $\ptl(\min(L)) \leq m + 1$. To see that $\ptl(\downarrow L) \leq m + 1$ we note that $\downarrow L$ coincide with $(\uparrow L) \setminus \min(L)$. □

**Remark 4.6.** The upper bound in Theorem 4.5 is quite good: for any $k, \eta \geq 1$, the language $L = \{ u_k \}$ has $\ptl(L) = n = k\eta + 1$ so that Eq. (2) gives $m = f_k(n) < (\eta + 2)^k$ and we obtain $\ptl(\uparrow L) < (\eta + 2)^k$ with Theorem 4.5. On the other hand Eq. (4) shows $\ptl(\downarrow L) \geq \eta^k + \eta^{k-1}$.

5 PT-level of downward closures

In this section we prove the following result.

**Theorem 5.1 (Downward closures).** Suppose $L \subseteq A^*$ is $n$-PT and $|A| = k$. Let $m = f_k(n)$. Then $\downarrow L$ and $\uparrow L$ are $(k + 1)(m + 1)$-PT.

While piecewise-testability follows immediately from the languages being downward-closed, bounding their PT-level is much harder. We relegate some technical proofs to Appendix C.

Recall Lemma 4.1.2 stating that, given two words $u, v$ and a letter $a \in A$, $uv \sim_n uv$ entails $uv \sim_n u\ell v$ for all $\ell \in \mathbb{N}$. We express this property as “$uv \in [uv]_n$ implies $ua^*v \subseteq [uv]_n$” and call it a pumping property of PT classes. To prove Theorem 5.1 we first establish more general pumping properties.

**Lemma 5.2.** If $uB^*C^*B^*v \subseteq [uv]_n$, where $B, C \subseteq A$ are subalphabets then $u(B \cup C)^*v \subseteq [uv]_n$.

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

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

**Lemma 5.3.** Suppose $L_1B_1^*B_2^* \cdots B_\ell^*L_2 \subseteq [u]_n$ for some languages $L_1, L_2 \subseteq A^*$ and subalphabets $B_1, B_2, \ldots, B_\ell \subseteq A$ with $\ell \geq 3$. If $a \in B_1 \cap B_\ell$ then, letting $B_i = B_i \cup \{ a \}$, $L_1B_1^*B_2^* \cdots B_\ell^*L_2 \subseteq [u]_n$.

**Proof.** This is Lemma C.4 in Appendix C. □
The following is a combinatorial lemma which will be useful later:

**Lemma 5.4.** Suppose $A$ is a finite set and $E_1, E_2, \ldots, E_\ell$ are subsets of $A$ such that the following hold:
- for all $1 \leq i < \ell$, $E_i \not\subseteq E_{i+1}$ and $E_{i+1} \not\subseteq E_i$ ;
- for all $a \in A$ and $1 \leq i < j \leq \ell$, if $a \in E_i \cap E_j$, then $a \in E_k$ for all $i \leq k \leq j$.

Then $\ell \leq |A|$.

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

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

**Lemma 5.5.** Let $L \subseteq A^*$ be $n$-PT. Let $k = |A|$ and $m = f_k(n)$. For every $u \in L$ there is a D-product $P_u$ of length $\ell \leq mk + m + k$ such that $u \in P_u \subseteq L$.

**Proof idea.** Assume $u \in L$. By Proposition 4.2 there is $v \subseteq u$ with $v \sim_n u$ and $|v| \leq m$. We identify occurrences of letters in $u$ which correspond to $v$. All other occurrences of letters can be pumped by Lemma 4.1.2 (that is, each $a$ can be replaced by $a^*$ while remaining in $[u]_n$). We then apply Lemma 5.3 to sequences of Kleene stars, and subsequently simplify whenever a $B^*$ and a $B^*$ appear adjacent with $B \subseteq B'$ or $B' \subseteq B$. The resulting sequences of Kleene stars satisfy the hypothesis of Lemma 5.4, which bounds their lengths. See Lemma C.5 in Appendix C for details. \qed

D-products are in general not PT, but we are interested in the following property:

**Lemma 5.6.** If $P$ is a D-product of length $\ell$, $ptl(\downarrow P) \leq \ell + 1$ and $ptl(\downarrow < P) \leq \ell + 1$.

**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 the canonical DFA for $P'$ has at most $\ell + 2$ states. Indeed, any residual $P'/w$ of $P'$ is either the empty language $\emptyset$, or corresponds to a suffix $P''$ of $P'$. This is easily shown by induction on the length of suffixes, starting with $\varepsilon/a = \emptyset$ for the last suffix (the empty product), and using

\[
[\downarrow (a + \varepsilon)P']/b = \begin{cases} P'' & \text{if } b = a, \\ P''/b & \text{otherwise}, \end{cases}
\]

\[
[B^*P'']/a = \begin{cases} B^*P'' & \text{if } a \in B, \\ P''/b & \text{otherwise}. \end{cases}
\]

We now observe that the depth of the DFA is at most $\ell + 1$ and apply Theorems 1 and 2 from [14] to deduce that $ptl(\downarrow P) \leq \ell + 1$. 


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

We may now conclude:

**Proof of Theorem 5.1.** Using notation from Lemma 5.5, from $u \in P_u \subseteq L$ we deduce $L = \bigcup_{u \in L} P_u$. Since every $P_u$ has length at most $km + k + m$, the above is a finite union and we may write $ptl(L) = \max_{u \in L} ptl(P_u)$. Now Lemma 5.6 entails $ptl(L) \leq km + k + m + 1 = (k + 1)(m + 1)$ for every $u \in L$.

The same reasoning applies for $\downarrow_L$.

**Remark 5.7.** The upper bound in Theorem 5.1 is quite good: for any $k, \eta \geq 1$, the language $L = \{u_k\}$ from Proposition 4.4 has $ptl(L) = n = k\eta + 1$ so that Theorem 5.1 gives $ptl(L) < (k + 1)(\eta + 2)^k$ while by Eq. (5) we know that $ptl(L) > \eta^k + \eta^{k-1}$.

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

In this section we prove the following result.

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

It is not too difficult to show the regularity of $I(L)$ when $L$ is regular, and this can be done using standard automata-theoretical techniques. Indeed, it can be shown that the incomparability relation $\perp$ is a rational relation [11].

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

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

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

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

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

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

The above examples illustrate our strategy for proving Theorem 6.1: if a language $L$ is totally ordered by $\sqsubseteq$ then $I(L) \cap L = \emptyset$, or equivalently $I(L) \sqsubseteq \downarrow_L$, so $I(L)$ is $n$-PT. But if $L$ is not totally ordered by $\sqsubseteq$, then $I(L)$ is not necessarily PT. For example, in $A = \{a, b\}$ the language

$$L = \{a^\ell b^\ell (\varepsilon + b) \mid \ell \in \mathbb{N}\}$$

is not totally ordered, since $a^\ell b^\ell (\varepsilon + b) > b^{\ell+1} (\varepsilon + b)$ for $\ell > 0$. Therefore $I(L)$ is not PT.
$A^* \subseteq L$. Similarly, if $L$ contains at least two words having same length $\ell$ then $I(L)$ contains all words of length $\ell$.

We now proceed with a more formal proof. For technical convenience we will analyse $C(L)$ rather than $I(L)$, knowing that these two languages have the same PT-level. Recall that $C(L) = \{ u \in A^* \mid L \subseteq \uparrow u \cup \downarrow u \}$ is the complement of $I(L)$.

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

For the rest of this section, we consider a fixed $n \geq 1$ and let $m = f_k(n)$. We start with a technical lemma.

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

**Proof idea.** Since $|u| = |v| = |w| - 1$, $w$ must be some $w_0 a_1 w_1 a_2 w_2$ with $a_1, a_2 \in A$ such that $u = w_0 a_1 w_1 w_2$ and $v = w_0 a_2 a_2 w_2$. We claim that $w' := w_0 w_1 a_1 w_2$ witnesses the lemma. Since $u \neq v$, we have $a_1 w_1 \neq a_2 w_2$, and thus $w \neq w'$. There remains to show that $w \sim_n w'$: this is done by a standard case analysis, see Appendix D. 

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

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

**Proof.** Suppose that $T$ contains two words of length $p$. Then, by Lemma 4.1.1, $T$ contains at least one word of length $p + 1$. Applying Lemma 6.2, we deduce that $T$ contains at least two words of length $p + 1$.

Populous layers also propagate downwards in the following sense:

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

**Proof.** Let $q$ be the smallest layer such that $T \cap A^q$ is populous. If $q = p + 1$ we are done, and similarly if $q = p$ (Lemma 6.3). So assume $q \geq p + 2$. For all $\ell$ with $p \leq \ell < q$, the layers $T \cap A^\ell$ are nonempty (by Lemma 4.1.1) hence singular. Further, Lemma 4.1 tells us the form of the words in these layers: $T \cap A^{q-\ell} = \{ uv, uav, \ldots, uav^{q-\ell} v \}$ for some $u, v \in A^*$ and $\ell \in A$.

We now turn to $T \cap A^q$. This populous layer contains some word $w \neq uav^{q-\ell} v$. By Theorem 6.2.9 of [19], all minimal (with respect to $\subseteq$) words of $T$ have the same length, hence $w$ is not minimal in $T$, and is obtained by inserting a single letter in $uav^{q-\ell} v$. Define a word $s$ as follows, depending on $w$:

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


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

We now handle a special case:

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

**Proof.** Assume $T$ is not linearly ordered by $\subseteq$ and pick $u, v \in T$ with $u \nsubseteq v$ and $|u| \leq |v|$. Let $q \defeq |v|$. By Lemma 4.1.1, there exists $w \in T$ such that $u \subseteq_n w$ and $v \subseteq_n w$. Along the sequence of words witnessing $u \subseteq_n w$, there exists a word $v'$ having length $q$. Furthermore, $v' \neq v$ since $u \nsubseteq v$ and $u \subseteq v'$. Thus $T \cap A^{\sim q}$ is populous. Since by Proposition 4.2 the shortest word in $T$ has length at most $m$, we conclude by Lemmas 6.3 and 6.4 that $T \cap A^{\sim p}$ is populous for every $p > m$. Thus $C(T) \subseteq A^{\leq m}$.

We now consider the general case:

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

**Proof.** Recall that $T$ is a singleton or is infinite (Lemma 4.1.3). We consider three cases.

- Suppose $T$ is a singleton, $T = \{u\}$. By Proposition 4.2, $|u| \leq m$. Then $|u|$ is $m$-PT, and $\uparrow u$ is $(m + 1)$-PT. Thus $I(\{u\}) = A^* \setminus (\uparrow u \cup \downarrow u)$ is $(m + 1)$-PT.
- Suppose $T$ is not a total order under $\subseteq$. Then by Lemma 6.5, $C(T) \subseteq A^{\leq m}$, so $C(T)$ is $(m + 1)$-PT, and so is $I(T)$.
- Suppose $T$ is infinite and a total order under $\subseteq$. Let $p$ be the length of the shortest word in $T$. By Proposition 4.2, $p \leq m$. Since $T$ is infinite, by Lemma 4.1.1 $T \cap A^{\sim q}$ is nonempty for every $q \geq p$. Since $T$ is a total order under $\subseteq$, none of these $T \cap A^{\sim q}$ is populous, hence they are all singular. Therefore $C(T) \cap A^{\sim p} = T$. It remains to describe $C(T) \cap A^{\leq p}$, and this is $\downarrow u_0$, where $u_0$ is the unique word of length $p$ in $T$. Thus $C(T) = T \cup \downarrow u_0$ is $(p + 1)$-PT, hence also $(m + 1)$-PT, and $I(T)$ too is $(m + 1)$-PT.

We may now conclude:

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

**Remark 6.7.** The upper bound in Theorem 6.1 is quite good: for any $k, \eta \geq 1$, the language $L = \{u_k\}$ from Proposition 4.4 has $ptl(L) = n = kn + 1$ so that Theorem 6.1 gives $ptl(I(L)) \leq (\eta + 2)^k$ while by Eq. (5) we know that $ptl(I(L)) > \eta^k + \eta^{k-1}$. 

7 Concluding remarks

We proved a $3^\text{–EXPTIME}$ upper bound on the complexity of $\text{FO}^2(A^*, \sqsubseteq)$, the two-variable fragment of the logic of subsequences. Some questions remain open: how can one narrow the gap between $\text{PSPACE}$ and $3^\text{–EXPTIME}$ for the basic logic, and what is the complexity of the $\text{FO}^2$ logic enriched with regular predicates?

Our result is obtained via a careful analysis of the PT-level of languages built by the decision procedure for $\text{FO}^2(A^*, \sqsubseteq)$. Indeed, we developed several new techniques for bounding the PT-levels of languages constructed as pre-images of the subword ordering or the associated incomparability relation. We believe that the PT hierarchy can be used more generally as an effective measure of descriptive complexity. This research program raises many interesting questions, such as connecting PT-levels and other measures, narrowing the gaps remaining in our Theorems 4.5, 5.1, and 6.1, and enriching the known collection of PT-preservation operations.

References

In this section we prove Proposition 4.2, i.e., the claim that for all $u \in A^n$ and $n \in \mathbb{N}$ there exists some $v \subseteq u$ with $|v| \leq f_k(n)$ and such that $v \leq_n u$, where $k = |A|$. This is an extension of results developed in [8].

Assume a fixed $k$-letter alphabet $A$.

We say that a word $x$ is rich if all the $k$ letters of $A$ occur in it, and that it is poor otherwise. For $\ell > 0$, we further say that $x$ is $\ell$-rich if it can be written as a concatenation of $\ell$ rich factors (by extension “$x$ is 0-rich” means that $x$ is poor). The richness of $x$ is the largest $\ell \in \mathbb{N}$ such that $x$ is $\ell$-rich. Note that having $|x|_a \geq \ell$ for all letters $a \in A$ does not imply that $x$ is $\ell$-rich. We shall use the following easy result:

**Lemma A.1.** If $x_1$ and $x_2$ are respectively $\ell_1$-rich and $\ell_2$-rich, then $y \sim_n y'$ implies $x_1yx_2 \sim_{\ell_1+n+\ell_2} x_1y'x_2$.

**Proof.** A subword $u$ of $x_1yx_2$ can be decomposed as $u = u_1vu_2$ where $u_1$ is the largest prefix of $u$ that is a subword of $x$ and $u_2$ is the largest suffix of the remaining $u_1^{-1}$ that is a subword of $x_2$. Thus $v \subseteq y$ since $u \subseteq x_1yx_2$. Now, since $x_1$ is $\ell_1$-rich, $|u_1| \geq \ell_1$ (unless $u$ is too short), and similarly $|u_2| \geq \ell_2$ (unless ...). Finally $|v| \leq n$ when $|u| \leq \ell_1 + n + \ell_2$, and then $v \subseteq y'$ since $y \sim_n y'$, entailing $u \subseteq x_1y'x_2$. A symmetrical reasoning shows that subwords of $x_1y'x_2$ of length $\leq \ell_1 + n + \ell_2$ are subwords of $x_1yx_2$ and we are done. \[\square\]
The rich factorization of $x \in A^*$ is the decomposition $x = x_1a_1 \cdots x_ma_my$ obtained in the following way: if $x$ is poor, we let $m = 0$ and $y = x$; otherwise $x$ is rich, we let $x_1a_1$ (with $a_1 \in A$) be the shortest prefix of $x$ that is rich, write $x = x_1a_1x'$ and let $x_2a_2 \cdots x_ma_my$ be the rich factorization of the remaining suffix $x'$. By construction $m$ is the richness of $x$. E.g., assuming $k = 3$ and $A = \{a,b,c\}$, the following is a rich factorization with $m = 2$:

$$
\begin{array}{c}
\hline
x & \hline
bbaabb & c \cdot cccaa \cdot b \cdot bbbaa \\
\hline
\end{array}
$$

Note that, by definition, $x_1, \ldots, x_m$ and $y$ are poor.

**Lemma A.2.** Consider two words $x, x'$ of richness $m$ and with rich factorizations $x = x_1a_1 \cdots x_ma_my$ and $x' = x'_1a_1' \cdots x'_ma_my'$. Suppose that $y \sim_n y'$ and that $x_i \sim_{n+1} x'_i$ for all $i = 1, \ldots, m$. Then $x \sim_{n+m} x'$.

**Proof.** By repeatedly using Lemma A.1, one shows

\[
x_1a_1x_2a_2 \cdots x_ma_my \sim_{n+m} x'_1a_1x'_2a_2 \cdots x'_ma_my
\]

\[
\sim_{n+m} x'_1a_1x'_2a_2 \cdots x'_ma_my
\]

\[
\vdots
\]

\[
\sim_{n+m} x'_1a_1x'_2a_2 \cdots x'_ma_my'
\]

using the fact that each factor $x_i a_i$ is rich.

**Proof (of Prop. 4.2).** By induction on $k$.

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

For the inductive case where $k > 1$ we consider the rich factorization $u = u_1a_1u_2a_2 \cdots u_ma_my$ of $u$. Let $n' = \max(n + 1 - m, 1)$. Every $u_i$ is a word on the subalphabet $A \setminus \{a_1\}$. Hence by induction hypothesis there exist $v_i \subseteq u_i$ with $|v_i| \leq f_{k-1}(n')$ and $v_i \sim_{n'} u_i$, entailing $u_ia_i \sim_{n'} v_ia_i$. Similarly, the induction hypothesis entails the existence of some $v' \subseteq u'$ with $v' \sim_{n'-1} u'$ and $|v'| \leq f_{k-1}(n' - 1)$. Note that in these inductive steps we use a length bound obtained with $f_{k-1}$ by profiting from 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. First, if $m < n$, we let $v = v_1a_1 \cdots v_ma_my'$, so that $v \subseteq u$ and $|v| \leq mf_{k-1}(n') + m + f_{k-1}(n' - 1)$. We deduce $|v| \leq f_k(n)$ using Eq. (1) and since $n' = n + 1 - m$. That $v \sim_n u$ is an application of Lemma A.2.

If $m \geq n$, then $u$ is $n$-rich and any word of a length $n$ is a subword of $u$. It is enough to take $v = u_1a_1 \cdots u_na_n$ and one obtains $u \sim_n v$. Finally, we also have $v \sim_n u$ because for every $w$ such that $v \subseteq w \subseteq u$, $v \sim_n w \sim_n u$. 

\[\square\]
B Proof for claim from Section 4.1

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

$$
(\bigwedge_{v \in P_k} v \subseteq u) \land (\bigwedge_{w \in N_k} w \not\subseteq u) \iff u = u_k. \quad (3)
$$

Proof. By induction on $k$. For $k = 0$, $A_0$ is empty and there is only one word in $A_0^*$, namely $u = u_0 = \varepsilon$. It satisfies the positive constraint $u_0 \subseteq u$ and there are no negative constraints in $N_0$.

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

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

C Proofs for Section 5 and Theorem 5.1

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

Proof sketch. Recall that $w_1 \sim_n w_2$ and $w_1 \subseteq w_2$ implies $w_1 \sim_n w'$ for all $w_1 \subseteq w' \subseteq w_2$. \qed

Lemma C.2 (5.2). If $uB^*C^*B^*v \subseteq [uv]_n$, where $B, C \subseteq A$ are subalphabets then $u(B \cup C)^*v \subseteq [uv]_n$.

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

Assume therefore that $m > 1$ and write $w$ as $w = xyz$ with $x \in B^*C^*$, $y \in B^*(C^*B^*)^{m-2}$, and $z \in C^*B^*$. If $s \subseteq uvw = uxzyv$ then $s$ can be factored as $s = s_0s_x s_y s_z s_0$ with each factor $s_\cdot$ a subword of the corresponding factor of $uvw$. Let $s' \defeq s_0s_x s_x s_0$ so that $s' \subseteq uxxzyv$. Note that $xz \in B^*C^*B^*$ hence $s' \subseteq uxxzyv$ entails $s' \subseteq uv$ by assumption. Thus either $s_0s_x \subseteq u$ or $s_x s_0 \subseteq v$.

In the first case, $s = s_0s_x s_y s_z s_0 \subseteq uyyzy$ and since $yz \in B^*(C^*B^*)^{m-1}$ the induction hypothesis applies and yields $s \subseteq uv$.

In the second case a symmetrical reasoning applies. \qed

Lemma C.3. If $uB^*C^*LD^*B^*v \subseteq [uv]_n$, where $B, C, D \subseteq A$ are subalphabets and $L \subseteq A^*$ is any language then $u(B \cup C)^*L(B \cup D)^*v \subseteq [uv]_n$. 
Proof. We assume that $L \neq \emptyset$ (otherwise the result holds trivially) so that $uB^*C^*LD^*B^*v \subseteq [uv]_n$ entails $uB^*C^*B^*v \subseteq [uv]_n$ (by Lemma C.1), hence $u(B \cup C)^*v \subseteq [uv]_n$ (by Lemma C.2).

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

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

Since $s \subseteq uvw = uxyzv$ there is a factorization $s = u_sw_xu_yzu_zv$ of $s$ with each factor $s_i$ embedding in the corresponding factor of $uxyzv$. Let now $s' \triangleq s_uu_xu_ys_zv$; this word satisfies $s' \subseteq uxyzv$ and $|s'| \leq n$. Now $uxzv \in uB^*C^*L(D^*B^*)^\ell v$, so that we may apply the induction hypothesis and deduce $s' \subseteq uxyzv$ from $s' \subseteq uxyzv$. Thus either $s_uu_xu \subseteq u$ or $s_zv \subseteq v$.

If $s_uu_xu \subseteq u$ we deduce

$$s = s_uu_xu_s_ys_zv \subseteq uyzv. \tag{7}$$

Now $yz \in (B^*C^*)^{k-1}L(D^*B^*)^\ell$ so that we can apply the induction hypothesis and deduce $s \subseteq uv$ from (7).

If $s_zv \subseteq v$ we deduce

$$s = s_uu_xu_s_ys_zv \subseteq uxyv. \tag{8}$$

Now $xy \in (B^*C^*)^k$ so that $uxyv \in [uv]_n$ as we observed at the beginning. Thus from (8) we deduce $s \subseteq uv$. \hfill \Box

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

**Lemma C.4 (5.3).** Suppose $L_1B_1^*B_2^* \cdots B_\ell^*L_2 \subseteq [u]_n$ for some languages $L_1$, $L_2 \subseteq A^*$ and subalphabets $B_1$, $B_2$, $B_3$, $B_\ell \subseteq A$ with $\ell \geq 3$. If $a \in B_1 \cap B_\ell$ then, letting $B'_\ell = B_\ell \cup \{a\}$, $L_1B_1^*B_2^* \cdots B_{\ell-1}^*B'_\ell L_2 \subseteq [u]_n$.

**Proof.** By induction on $\ell$. Write $L_1B_1^*B_2^* \cdots B_\ell^*L_2$ as $L_1B_1^*a^*B_2^* \cdots B_{\ell-1}^*a^*B_{\ell}^*L_2$. For every $u_1 \in L_1B_1^*$ and $u_2 \in B_2^*L_2$, we have $u_1a^*B_2^* \cdots B_{\ell-1}^*a^*u_2 \subseteq [u_1u_2]_n = [u]_n$. Now by Lemma C.3, we deduce $u_1B_2^*B_3^* \cdots B_{\ell-2}^*B_{\ell-1}^*a^*u_2 \subseteq [u]_n$, and then $u_1B_{\ell}^*B_{\ell-1}^* \cdots B_{\ell-2}^*B_{\ell-1}^*u_2 \subseteq [u]_n$ by the induction hypothesis. Since this applies to all $u_1 \in L_1B_1^* = L_1B_1^*$ and $u_2 \in B_2^*L_2 = B_{\ell}^*L_2$, we have proven the lemma. \hfill \Box

**Lemma C.5 (5.5).** Let $L \subseteq A^*$ be $n$-PT. Let $k = |A|$ and $m = f_k(n)$. For every $u \in L$ there is a $D$-product $P_u$ of length $\ell \leq mk + m + k$ such that $u \in P_u \subseteq L$.

In the above statement (and below) we abuse notation and let $P$ denote both a regular expression and the language (a subset of $A^*$) it denotes.
Proof. Assume \( u \in L \). By Proposition 4.2 there is \( v \subseteq u \) with \( v \sim_n u \) and \( |v| \leq m \). Assume \( v = a_1 \cdots a_{|v|} \) so that \( u \) has the form

\[
u = b_{0,1} \cdots b_{0,p_0} a_1 b_{1,1} \cdots b_{1,p_1} a_2 \cdots a_{|v|} b_{|v|,1} \cdots b_{|v|,p_{|v|}}\]

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

By the pumping property (Lemma 4.1.2), we deduce that \( P = \prod_{i=0}^{|v|} (a_i \prod_{j=1}^{p_i} (b_{i,j})^*) \) is a D-product with \( u \in P \subseteq L \).

We now modify this product to take advantage of the more general pumping property. Let \( P' = \prod_{i=0}^{|v|} (a_i \prod_{j=1}^{p_i} B_{i,j}^*) \) where \( B_{i,j} = \{b_{i,j}\} \cup \{a \in A \mid \exists 1 \leq j_1 < j < j_2 \leq p_i : a = b_{i,j_1} = b_{i,j_2}\} \). That is, every subalphabet \( \{b_{i,j}\} \) in \( P \) is completed with any letter that appears both before and after \( b_{i,j} \) in the same \( i \)-th segment \( B_{i,1}^* \cdots B_{i,p_i}^* \). Now Lemma C.4 ensures that \( P' \subseteq L \) (and we still have \( u \in P' \) since \( P \subseteq P' \)).

We now simplify \( P' \) by repeatedly replacing a factor \( B^* B'^* \) where \( B \subseteq B' \) (or \( B' \subseteq B \)) by \( B^* \) (or \( B'^* \)). This does not change the language denoted by \( P' \). When no more simplification is possible, we let \( P_u = \prod_{i=0}^{|v|} (a_i \prod_{j=1}^{p_i} C_{i,j}^*) \) denote the simplified D-product. We have that for all \( i \in \{0, \ldots, |v|\} \), the sequence of sets \( C_{i,1}, \ldots, C_{i,p_i} \) satisfies the hypothesis of Lemma 5.4, and thus \( \ell_i \leq k = |A| \). This entails that \( P_u \) has length bounded by \((m+1)(k+1)-1\) (recall that \( a_0 = \varepsilon \)), i.e. by \( mk + m + k \).

D Proof that \( w \sim_n w' \) for Lemma 6.2

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

So let \( s \subseteq w' \) with \( |s| \leq n \). Factorize \( s \) as \( s = v_0v_1s'v_2 \) as follows:

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

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

Claim. \( v_1a_2v_2 \subseteq a_1w_1w_2 \).

Proof. The claim asserts that a certain suffix of \( t \) is a subword of a certain suffix of \( u \). We know that \( t \subseteq u \), i.e., \( v_0a_1v_1a_2v_2 \subseteq w_0a_1w_1w_2 \). Hence if \( v_1a_2v_2 \subseteq a_1w_1w_2 \), then \( v_1a_1\alpha \subseteq w_0 \) for some nonempty prefix \( \alpha \) of \( v_1a_2v_2 \). But this contradicts the definition of \( v_0 \).
Now since $v_1 a_2 v_2 \subseteq a_1 w_1 w_2$, we have $v_1 a_2 \subseteq a_1 w_1$ or $a_2 v_2 \subseteq w_2$. Combining this with $v_1 \subseteq w_1$ and $v_2 \subseteq w_2$, we get $v_1 a_2 a_2 v_2 \subseteq a_1 w_1 a_2 w_2$. Finally, this along with $v_0 \subseteq w_0$ gives $s \subseteq w$ as needed.