

Generalized Core Spanner Inexpressibility via Ehrenfeucht-Fraïssé Games for FC

SAM M. THOMPSON, Loughborough University, UK

DOMINIK D. FREYDENBERGER, Loughborough University, UK

Despite considerable research on document spanners, little is known about the expressive power of generalized core spanners. In this paper, we use Ehrenfeucht-Fraïssé games to obtain general inexpressibility lemmas for the logic FC (a finite model variant of the theory of concatenation). Applying these lemmas give inexpressibility results for FC that we lift to generalized core spanners. In particular, we give several relations that cannot be selected by generalized core spanners, thus demonstrating the effectiveness of the inexpressibility lemmas. As an immediate consequence, we also gain new insights into the expressive power of core spanners.

CCS Concepts: • **Theory of computation** → **Logic and databases**; *Finite Model Theory*.

Additional Key Words and Phrases: Information extraction, document spanners, finite model theory, expressive power, Ehrenfeucht-Fraïssé games

ACM Reference Format:

Sam M. Thompson and Dominik D. Freydenberger. 2024. Generalized Core Spanner Inexpressibility via Ehrenfeucht-Fraïssé Games for FC. *Proc. ACM Manag. Data* 2, 2 (PODS), Article 80 (May 2024), 18 pages. <https://doi.org/10.1145/3651143>

1 INTRODUCTION

Ehrenfeucht-Fraïssé games are a fundamental tool for establishing inexpressibility results in finite model theory. In this paper, we develop techniques to use Ehrenfeucht-Fraïssé games for the logic FC and use these to gain insights into the expressive power of generalized core spanners.

Document Spanners. SystemT is a rule-based information extraction system, developed by IBM, which includes a declarative query language called Annotation Query Language (AQL) – see [3, 19, 20]. Fagin, Kimelfeld, Reiss, and Vansummeren [6] developed a formal framework for information extraction called *document spanners* (or *spanners*), which captures the core functionality of AQL.

One can define the process of querying a text document with a spanner as a two-step process. First, so-called *extractors* obtain relations of intervals (or *spans*) from the text. For the purposes of this introduction, we assume these extractors to be *regex formulas*, which are regular expressions with capture variables. For example, if someone wanted to find all occurrences of common misspellings in a text document, they could consider the regex formula

$$\gamma(x) := \Sigma^* \cdot x\{\text{acheive} \vee \text{begining} \vee \dots \vee \text{wether}\} \cdot \Sigma^*.$$

Then, γ extracts a unary relation of spans within the given text document where one of the misspellings occur.

Secondly, the extracted relations are combined using a relational algebra. This paper is concerned with the class of *generalized core spanners*, which allow for $\cup$ (union), π (projection), $\bowtie$ (natural

Authors' addresses: Sam M. Thompson, s.thompson4@lboro.ac.uk, Loughborough University, Loughborough, UK; Dominik D. Freydenberger, D.D.Freydenberger@lboro.ac.uk, Loughborough University, Loughborough, UK.



This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2024 Copyright held by the owner/author(s).

ACM 2836-6573/2024/5-ART80

<https://doi.org/10.1145/3651143>

join), $\setminus$ (difference), and $\zeta^=$ (equality selection). While union, projection, natural join, and difference are defined as one might expect (assuming prior knowledge of relational algebra), equality selection is more text specific. The equality selection $\zeta_{x,y}^=$ only chooses those tuples for which the variables x and y are mapped to spans that represent the same factor (at potential different locations).

A word relation $R \subseteq (\Sigma^*)^r$, for some $r \in \mathbb{N}$, is *selectable* by generalized core spanners if adding an R selection operator (that is, ζ^R) to our relational algebra does not increase the expressive power, see [6] for more details on selectable relations. One fundamental question regarding generalized core spanners is what relations are selectable? For example, from Theorem 5.14 of [12], we know that length equality $\zeta_{x,y}^{\text{len}}$ (i.e., only keep those tuples for which the span that x is mapped to has the same length as the span that y is mapped to) cannot be expressed by generalized core spanners.

Ehrenfeucht-Fraïssé Games and FC. For relational first-order logic (and therefore relational algebra), one of the fundamental tools for inexpressibility are *Ehrenfeucht-Fraïssé games*; for example, see [4, 16, 21]. To use Ehrenfeucht-Fraïssé games in order to yield inexpressibility results for generalized core spanners, we need a suitable logic. For this task, we use FC; a logic that was introduced by Freydenberger and Peterfreund [12] as a finite model variant of the theory of concatenation. For this paper, we define this logic slightly differently to [12]. First, we treat every word $w \in \Sigma^*$ as a relational structure consisting of a universe that contains all factors of w , a ternary concatenation relation $R_\circ$ where $(x, y, z) \in R_\circ$ if and only if x, y , and z are all factors of w where $x = y \cdot z$, and constant symbols for every $a \in \Sigma$ and for the empty word ε . Then, FC is simply a first-order logic over these structures. As syntactic sugar, we use $(x \doteq y \cdot z)$ rather than $R_\circ(x, y, z)$ for atomic FC-formulas. As an example, consider the formula

$$\varphi := \forall z: (\neg(z \doteq \varepsilon \cdot \varepsilon) \rightarrow \neg \exists x, y: (x \doteq z \cdot y) \wedge (y \doteq z \cdot z)).$$

This formula states that for all factors z where $z = \varepsilon$ does not hold (all non-empty factors), we have that there does not exist a factor x where $x = zzz$. In other words, φ defines those words $w \in \Sigma^*$ that do not contain uuu where u is a non-empty word.

If we consider Ehrenfeucht-Fraïssé games over two structures that represent a word $w \in \Sigma^*$, we can gain inexpressibility results for FC. Then, using techniques from combinatorics on words (namely, *commutation*), we are able to lift these inexpressibility results to generalized core spanners.

Related Work. Fagin et al. [6] introduces three main classes of document spanners. The *regular spanners* (regex formulas extended with projection, union, and natural join) are perhaps the most intensively studied class with a sizeable amount of research on enumeration algorithms [2, 8, 29]. When it comes to expressive power, Fagin et al. [6] showed that regular spanners cannot go beyond *recognizable relations*. The *core spanners* extend regular spanners with equality selection, allowing for more expressive power at the cost of tractability, see e.g. [9, 10, 12]. With regards to their expressive power, [6] gave the *core-simplification lemma*, allowing one to simplify core spanners into a normal form in order to study their expressive power. Freydenberger [9] uses inexpressibility techniques from word equations (namely, results in Karhumäki, Mignosi, and Plandowski [18]) to obtain an inexpressibility lemma for core spanners, which yields relations that cannot be selected by core spanners.

When it comes to generalized core spanners, the techniques from core spanners do not work. As far as the authors are aware, only two inexpressibility results for generalized core spanners exist in literature: Peterfreund, ten Cate, Fagin, and Kimelfeld [26] showed that over a unary alphabet, only *semi-linear languages* can be expressed by generalized core spanners. Then, Freydenberger and Peterfreund [12] used the *Feferman-Vaught theorem* to show that the language $\{a^n b^n \mid n \in \mathbb{N}\}$ is not expressible by generalized core spanners. However, the use of the Feferman-Vaught theorem in [12] relies on the limited structure of the language $a^n b^n$, and it is unclear how to generalize this

proof technique. Note that from the data complexity of model checking FC-formulas, we know that it cannot express languages outside of LOGSPACE, see Proposition A.5 of [11]. While this observation might not be considered particularly enlightening from an inexpressibility point of view, we state this for completeness sake.

The expressive power of other classes of spanners (that is, apart from regular spanners, core spanners, and generalized core spanners) has also been considered. For example, see [25, 26, 28, 32].

Issues with Standard Techniques. Beyond toy examples, Ehrenfeucht-Fraïssé games are difficult to play, and usually require rather involved combinatorial arguments. Therefore, many techniques have been developed in order to achieve sufficient criteria that give Duplicator a winning strategy (see [5] for a survey). For example, *locality* is often used to show that Duplicator has a winning strategy. Unfortunately, these locality results often fail for *non-sparse structures* (for example, see Section 5.3 of [21]). The structures we look at in the present paper are very non-sparse, and therefore we cannot simply apply locality results in order to gain inexpressibility tools.

Another useful tool is the *Feferman-Vaught theorem* (for example, see [24]). Freydenberger and Peterfreund [12] used this result to prove that $\{a^n b^n \mid n \in \mathbb{N}\}$ is not an FC-language, which then implies that length selection is unattainable for generalized core spanners. To this end, [12] defines FO[EQ], which extends first-order logic over a linear order with symbol predicates with a built-in equality relation (that is, the set of 4-tuples of positions where the factor defined by the first two components equals the factor defined by the second two components). Then, we can decompose these structures that represent words of the form $a^n b^m$ into two substructures (one for a^n and one for b^m). Finally, the Feferman-Vaught theorem can be invoked to prove that $\{a^n b^n \mid n \in \mathbb{N}\}$ is not an FC language. However, this proof idea does not generalize beyond words which can be decomposed into disjoint sections (at least not without a sizeable amount of extra machinery).

Structure of the Paper. Section 2 gives notational conventions and definitions that are used throughout this article. In Section 3, we define Ehrenfeucht-Fraïssé games for FC, and give some basic results. The main technical contributions are given in Section 4, where we work towards the *Fooling Lemma* for FC – a general tool for showing a language cannot be expressed in FC. On the way to the Fooling Lemma, we give two important lemmas called the *Pseudo-Congruence Lemma* and the *Primitive Power Lemma*. We then use our inexpressibility lemmas in Section 5 to obtain various relations that are not selectable by generalized core spanners. Due to space constraints, most of the longer proofs have been omitted (a version that includes proofs is available at [31]).

2 PRELIMINARIES

Let $\mathbb{N} := \{0, 1, 2, \dots\}$ and let $\mathbb{N}_+ := \mathbb{N} \setminus \{0\}$ where $\setminus$ denotes set difference. For $n \geq 1$, we use $[n]$ for $\{1, 2, \dots, n\}$. The cardinality of a set S is denoted by $|S|$. For a vector $\vec{a} \in A^k$ for some set A and $k \in \mathbb{N}$, we write $x \in \vec{a}$ to denote that x is a component of $\vec{a}$.

We use Σ for a fixed and finite alphabet of terminal symbols and we use Ξ for a countably infinite set of variables. If $w = w_1 \cdot w_2 \cdot w_3$ where $w, w_1, w_2, w_3 \in \Sigma^*$, then we call w_1 a prefix of w , w_2 a factor w , and w_3 a suffix of w . If $w_1 \neq w$, then w_1 is a *strict prefix* of w ; and likewise, if $w_3 \neq w$, then w_3 is a *strict suffix* of w . We denote that w_2 is a *factor* of w by $w_2 \sqsubseteq w$ and if $w_2 \neq w$ also holds, then $w_2 \subset w$. For $w \in \Sigma^*$, we write the set of all factors of w as $\text{Facs}(w) := \{u \in \Sigma^* \mid u \sqsubseteq w\}$. We use $|w|$ to denote the length of a word $w \in \Sigma^*$, and for some $a \in \Sigma$, we use $|w|_a$ to denote the number of occurrences of a within w . Let ε denote the empty word. For $w \in \Sigma^*$ and $k \in \mathbb{N}$, let $w^k \in \Sigma^*$ denote the word that consists of k repetitions of w . We always assume $w^0 = \varepsilon$.

A word $w \in \Sigma^*$ is called *imprimitive* if $w = z^k$ for some $z \in \Sigma^*$ and $k > 1$. We always assume ε to be imprimitive. If $w \in \Sigma^+$ is not imprimitive, then w is *primitive*. That is, $w \in \Sigma^+$ is primitive if for all $z \in \Sigma^+$, we have that $w = z^m$ implies $m = 1$.

The logic FC. We introduce a definition of FC that is slightly more technical than the original one from [12] and more suitable for defining games. The differences, and why they do not affect the expressive power, are discussed at the end of this section.

For the purposes of this paper, we consider one fixed signature $\tau_\Sigma := \{R_o, a_1, \dots, a_m, \varepsilon\}$ for every terminal alphabet $\Sigma := \{a_1, \dots, a_m\}$, where R_o is a ternary relation symbol and where ε and a_i for every $i \in [m]$ are constant symbols. Given a word $w \in \Sigma^*$, let $\mathfrak{A}_w := (A, R_o^{\mathfrak{A}_w}, a_1^{\mathfrak{A}_w}, \dots, a_m^{\mathfrak{A}_w}, \varepsilon^{\mathfrak{A}_w})$ be the τ_Σ -structure that represents $w \in \Sigma^*$ as follows:

- $A := \text{Facs}(w) \cup \{\perp\}$ is the universe,
- $R_o^{\mathfrak{A}_w} := \{(a, b, c) \in \text{Facs}(w)^3 \mid a = b \cdot c\}$ is concatenation, restricted to factors of w ,
- $a_i^{\mathfrak{A}_w} := a$ if $|w|_a \geq 1$ and $a_i^{\mathfrak{A}_w} = \perp$ otherwise, and
- $\varepsilon^{\mathfrak{A}_w} := \varepsilon$.

Note that if $|w|_a = 0$, then $a_i^{\mathfrak{A}_w} = \perp$. One can think of $\perp$ as a null value; however, we usually deal with those words where $|w|_a \geq 1$ for all $a \in \Sigma$. Therefore, due to the nature of the constant symbols used in τ_Σ , we often do not distinguish between a constant symbol $a \in \tau_\Sigma$, and the terminal symbol $a \in \Sigma$. In other words, when the structure is clear from context, we use $a \in \Sigma$ rather than $a_i^{\mathfrak{A}_w} \in A$.

An FC-formula is a first-order formula where the atomic formulas are of the form $R_o(x, y, z)$ for variables or constants x, y , and z . As syntactic sugar, we use $(x \doteq y \cdot z)$ as FC atomic formulas, as we always interpret R_o as concatenation. More formally:

Definition 2.1. We define FC, the set of all FC-formulas, recursively as:

- If $x, y, z \in \Xi \cup \Sigma \cup \{\varepsilon\}$, then $(x \doteq y \cdot z) \in \text{FC}$,
- if $\varphi, \psi \in \text{FC}$, then $(\varphi \wedge \psi) \in \text{FC}$, $(\varphi \vee \psi) \in \text{FC}$, and $\neg\varphi \in \text{FC}$,
- if $\varphi \in \text{FC}$ and $x \in \Xi$, then $\forall x: \varphi \in \text{FC}$ and $\exists x: \varphi \in \text{FC}$.

An interpretation $\mathcal{I} := (\mathfrak{A}_w, \sigma)$ consists of a τ_Σ -structure $\mathfrak{A}_w$ that represents some $w \in \Sigma^*$, and a mapping $\sigma: (\Sigma \cup \Xi \cup \{\varepsilon\}) \rightarrow \text{Facs}(w) \cup \{\perp\}$, where $\sigma(a) = a_i^{\mathfrak{A}_w}$ for each constant symbol $a \in \tau_\Sigma$. Furthermore, we always assume $\sigma(x) \neq \perp$ for all $x \in \Xi$.

Definition 2.2. For any $\varphi \in \text{FC}$, and any interpretation $\mathcal{I} := (\mathfrak{A}_w, \sigma)$, where $\mathfrak{A}_w$ represents $w \in \Sigma^*$, we define $\mathcal{I} \models \varphi$ as follows: Let $\mathcal{I} \models (x \doteq y \cdot z)$ if $(\sigma(x), \sigma(y), \sigma(z)) \in R_o^{\mathfrak{A}_w}$. For the quantifiers, we have that $\mathcal{I} \models \exists x: \varphi$ (or $\mathcal{I} \models \forall x: \varphi$) if $(\mathfrak{A}_w, \sigma_{x \rightarrow u}) \models \varphi$ holds for an (or all) $u \in \text{Facs}(w)$, where $\sigma_{x \rightarrow u}(x) = u$ and $\sigma_{x \rightarrow u}(y) = \sigma(y)$ for all $y \in \Xi \setminus \{x\}$. Negation, disjunction, and conjunction are defined canonically.

We use the usual first-order logic definition of bound variables and free variables (see, for example, Section 2.1 of [21]). If φ is a sentence (that is, there are no free variables in φ), then we simply write $\mathfrak{A}_w \models \varphi$. Furthermore, for any $w \in \Sigma^*$, there is a unique τ_Σ -structure $\mathfrak{A}_w$ that represents w . We can therefore define the *language* of an FC-sentence.

Definition 2.3. The language defined by a sentence $\varphi \in \text{FC}$ is $\mathcal{L}(\varphi) := \{w \in \Sigma^* \mid \mathfrak{A}_w \models \varphi\}$. Let $\mathcal{L}(\text{FC})$ be the class of languages definable by an FC-sentence.

We write $\llbracket \varphi \rrbracket(w)$ to denote the set of all mappings σ such that $(\mathfrak{A}_w, \sigma) \models \varphi$, where $\mathfrak{A}_w$ is the τ_Σ -structure that represents $w \in \Sigma^*$. Note that for simplicity, we assume that for any $\sigma \in \llbracket \varphi \rrbracket(w)$, the domain of σ is exactly the free variables of φ .

For a word relation $R \subseteq (\Sigma^*)^k$, we say that R is *definable in FC* if there exists $\varphi_R \in \text{FC}$ with free variables $x_1, x_2, \dots, x_k$ such that for any $w \in \Sigma^*$, we have $\sigma \in \llbracket \varphi_R \rrbracket(w)$ if and only if

$$(\sigma(x_1), \sigma(x_2), \dots, \sigma(x_k)) \in R \cap \text{Facs}(w)^k.$$

We say that such a formula φ_R *defines* R .

Example 2.4. Consider the following FC-formula:

$$\varphi_w(x) := \neg \exists z_1, z_2: (((z_1 \dot{=} z_2 \cdot x) \vee (z_1 \dot{=} x \cdot z_2)) \wedge \neg(z_2 \dot{=} \varepsilon)).$$

This formula states that there is no factor that is a concatenation of $\sigma(x)$ and a non-empty word. Thus, for any $w \in \Sigma^*$, it must be that $\llbracket \varphi \rrbracket(w) = \{\sigma\}$ where $\sigma(x) = w$. Note that we use $(z_2 \dot{=} \varepsilon)$ as a shorthand for $(z_2 \dot{=} \varepsilon \cdot \varepsilon)$.

We can use $\varphi_w(x)$ to define the following sentence:

$$\varphi_{ww} := \exists x: \exists y: (\varphi_w(x) \wedge (x \dot{=} yy)).$$

It follows that $\mathfrak{A} \models \varphi_{ww}$ if and only if $\mathfrak{A}$ is a τ_Σ -structure that represents ww for some $w \in \Sigma^*$.

Consider $\varphi(x, y) := (x \dot{=} yy)$ which defines the relation $R_{\text{copy}} := \{(u, v) \subseteq (\Sigma^*)^2 \mid u = vv\}$. Since we now know that R_{copy} can be defined in FC, we can use it as an additional atomic formula without changing the expressive power of FC. Furthermore, it is rather straightforward to generalize this relation to the FC-definable relation $R_{k\text{-copies}} := \{(u, v) \in (\Sigma^*)^2 \mid u = v^k\}$.

Before moving on, we first make a brief note on the difference between the definition of FC in this paper and the definition given in [12]. First, the definition of FC given in [12] allows for an arbitrarily large right-hand side. That is, atomic formulas of the form $x \dot{=} \alpha$ where $x \in \Xi$ and $\alpha \in (\Sigma \cup \Xi)^*$. Whereas, we use $(x \dot{=} y \cdot z)$ as atomic formulas, where $x, y, z \in \Xi \cup \Sigma \cup \{\varepsilon\}$. An arbitrarily large right-hand side is shorthand for a binary concatenation term (for example, see Freydenberger and Thompson [13]). The reason behind our choice is that we have a finite signature τ_Σ which more closely aligns with “traditional” finite model theory. The second difference is that we do not use a *universe variable* u to denote the “input word” (see [12] for more details). However, referring back to Example 2.4, we can simply use a subformula to simulate the behaviour of u . While there is also a small difference with the semantic definitions (such as the use of $\perp$), these differences are negligible and do not change the expressive power.

3 EHRENFUCHT-FRAÏSSÉ GAMES FOR FC.

In this section, we discuss the use of Ehrenfeucht-Fraïssé games for FC. Although broader definitions of the concepts defined here exist (see e.g. [4, 16, 21]), we tailor our definitions to the logic FC.

Definition 3.1 (Partial Isomorphism). Let $\mathfrak{A}_w$ and $\mathfrak{B}_v$ be two τ_Σ -structures that represent $w \in \Sigma^*$ and $v \in \Sigma^*$ respectively. Let $A := \text{Facs}(w) \cup \{\perp\}$ and $B := \text{Facs}(v) \cup \{\perp\}$ be the universes of these structures. For $\vec{a} = (a_1, \dots, a_n)$ and $\vec{b} = (b_1, \dots, b_n)$ where $a_i \in A$ and $b_i \in B$ for all $i \in [n]$, we say that $(\vec{a}, \vec{b})$ defines a *partial isomorphism* between $\mathfrak{A}_w$ and $\mathfrak{B}_v$ if:

- For every $i \in [n]$ and constant symbol $c \in \tau_\Sigma$, we have $a_i = c^{\mathfrak{A}_w}$ if and only if $b_i = c^{\mathfrak{B}_v}$,
- for every $i, j \in [n]$, we have that $a_i = a_j$ if and only if $b_i = b_j$, and
- for every $i, j, k \in [n]$, we have that $a_i = a_j \cdot a_k$ if and only if $b_i = b_j \cdot b_k$.

We now discuss Ehrenfeucht-Fraïssé games. This game is played between two players, called *Spoiler* and *Duplicator*. For our purposes, Ehrenfeucht-Fraïssé games are played on two τ_Σ -structures $\mathfrak{A}_w$ and $\mathfrak{B}_v$ that represent some $w \in \Sigma^*$ and $v \in \Sigma^*$ respectively. Let $A := \text{Facs}(w) \cup \{\perp\}$ and $B := \text{Facs}(v) \cup \{\perp\}$. The players play a fixed number (which we often denote with $k \in \mathbb{N}$) of rounds. For any $i \leq k$, the i -th round is played as follows:

Spoiler’s Move: Spoiler picks a structure $\mathfrak{A}_w$ or $\mathfrak{B}_v$, and then picks some element $a_i \in A$ if they chose $\mathfrak{A}_w$, or some $b_i \in B$ if they chose $\mathfrak{B}_v$.

Duplicator’s Move: Duplicator then responds by picking some element in the other structure.

For any τ_Σ -structure $\mathfrak{A}$ where $\Sigma := \{c_1, \dots, c_n\}$, we use $\vec{c}^{\mathfrak{A}}$ to denote $(c_1^{\mathfrak{A}}, \dots, c_n^{\mathfrak{A}}, \varepsilon^{\mathfrak{A}})$. Let $\mathfrak{A}_w$ and $\mathfrak{B}_v$ be two τ_Σ -structures that represent $w \in \Sigma^*$ and $v \in \Sigma^*$ respectively. For a k -round game $\mathcal{G}$

over the structures $\mathfrak{A}_w$ and $\mathfrak{B}_v$, let $a_i \in A$ and $b_i \in B$ be the elements chosen in round i for each $i \in [k]$. Furthermore, let $\vec{a} = (a_1, a_2, \dots, a_{k+|\Sigma|+1})$ and $\vec{b} = (b_1, b_2, \dots, b_{k+|\Sigma|+1})$ be tuples of the chosen elements, combined with the tuples $\vec{c}^{\mathfrak{A}_w}$ and $\vec{c}^{\mathfrak{B}_v}$ respectively. In other words, the tuples consisting of the last $|\Sigma| + 1$ components of $\vec{a}$ and $\vec{b}$ are $\vec{c}^{\mathfrak{A}_w}$ and $\vec{c}^{\mathfrak{B}_v}$ respectively. Then, Duplicator has won the k -round game if and only if $(\vec{a}, \vec{b})$ is a partial isomorphism between $\mathfrak{A}_w$ and $\mathfrak{B}_v$.

For a k -round game $\mathcal{G}$ over $\mathfrak{A}_w$ and $\mathfrak{B}_v$, we say that Duplicator has a *winning strategy* for $\mathcal{G}$ if and only if there is a way for Duplicator to play such that no matter what Spoiler chooses, Duplicator can win $\mathcal{G}$. Otherwise, Spoiler has a winning strategy for $\mathcal{G}$. If Duplicator has a winning strategy for $\mathcal{G}$, then we write $\mathfrak{A}_w \equiv_k \mathfrak{B}_v$. Since $\mathfrak{A}_w$ and $\mathfrak{B}_v$ are uniquely determined by $w, v \in \Sigma^*$, we can simply write $w \equiv_k v$ when convenient.

Example 3.2. Let $\Sigma = \{a\}$. For any $i \in \mathbb{N}_+$, Spoiler has a 2-round winning strategy on $\mathfrak{A}_w$ and $\mathfrak{B}_v$ where $w := a^{2i}$ and $v := a^{2i-1}$:

Round One. Spoiler chooses the structure $\mathfrak{A}_w$ and the factor $a_1 = a^{2i}$. Duplicator responds with some $b_1 \sqsubseteq v$.

Round Two. The choice for Spoiler in this round is broken into two cases, depending on Duplicator's choice in round one:

- (1) If Duplicator responds in round one with $b_1 = a^{2i-1}$, then Spoiler chooses the structure $\mathfrak{A}_w$ and $a_2 = a^i$. As there does not exist a factor $b_2 \sqsubseteq v$ such that $v = b_2 \cdot b_1$, Spoiler has won after two rounds.
- (2) If Duplicator responds in round one with $b_1 \neq a^{2i-1}$, then Spoiler chooses the structure $\mathfrak{B}_v$ and $b_2 = b_1 \cdot a$. Duplicator must respond with some factor a_2 , however a factor of the form $a_1 \cdot a$ does not exist.

For either of the cases given in round two, we have that $(a_1, a_2, a, \varepsilon)$ and $(b_1, b_2, a, \varepsilon)$ does not form a partial isomorphism and thus, Spoiler has a winning strategy for the two round game. Consequently, $a^{2i} \not\equiv_2 a^{2i-1}$ for any $i \in \mathbb{N}_+$.

Next, we connect Ehrenfeucht-Fraïssé games to FC. Each $\varphi \in \text{FC}$ has a so-called *quantifier rank*, denoted by the function $\text{qr}: \text{FC} \rightarrow \mathbb{N}$. Let $\varphi, \psi \in \text{FC}$. We define qr recursively as follows:

- If $\varphi \in \text{FC}$ is an atomic formula, then $\text{qr}(\varphi) := 0$,
- $\text{qr}(\neg\varphi) := \text{qr}(\varphi)$,
- $\text{qr}(\varphi \wedge \psi) = \text{qr}(\varphi \vee \psi) := \max(\text{qr}(\varphi), \text{qr}(\psi))$, and
- $\text{qr}(Qx: \varphi) := \text{qr}(\varphi) + 1$ for any $x \in \Xi$ and $Q \in \{\forall, \exists\}$.

Let $\text{FC}(k)$ denote the set of sentences $\varphi \in \text{FC}$ such that $\text{qr}(\varphi) \leq k$.

Note that the set of all τ_Σ -structures $\mathfrak{A}_w$ that represent words $w \in \Sigma^*$ are simply a subset of all structures over a signature that contains a ternary relational symbol and $|\Sigma| + 1$ constant symbols. Therefore, we can state an FC-version of the Ehrenfeucht-Fraïssé theorem (see e.g. [4, 16, 21] for the general version).

THEOREM 3.3. *Let $\mathfrak{A}_w, \mathfrak{B}_v$ be τ_Σ -structures that represent $w \in \Sigma^*$ and $v \in \Sigma^*$ respectively. Then, for any $k \in \mathbb{N}$, the following are equivalent:*

- (1) $\mathfrak{A}_w \equiv_k \mathfrak{B}_v$,
- (2) $\mathfrak{A}_w \models \varphi$ if and only if $\mathfrak{B}_v \models \varphi$ for all $\varphi \in \text{FC}(k)$.

Like how we can adapt the Ehrenfeucht-Fraïssé theorem to FC, we can also adapt the well-known result that Ehrenfeucht-Fraïssé games can completely characterize the expressive power of first-order logic (see, for example, Theorem 6.19 from [16]) to get the following:

LEMMA 3.4. *Let $L \subseteq \Sigma^*$. For every $k \in \mathbb{N}$, there exist $w \in L$ and $u \notin L$ where $w \equiv_k u$ if and only if L is not an FC-language.*

For our next step, we rely on an inexpressibility technique for languages over unary alphabets. Over a unary alphabet $\{a\}$, we can identify every unary word a^n with the natural number n , which allows us to treat languages $L \subseteq \{a\}^*$ as sets $S_L := \{n \in \mathbb{N} \mid a^n \in L\}$ of natural numbers. A set $S \subseteq \mathbb{N}$ is *linear* if there exist an $r \in \mathbb{N}$ and $m_0, \dots, m_r \in \mathbb{N}$ such that $S = \{m_0 + \sum_{i=1}^r m_i n_i \mid n_i \geq 0\}$. A set is *semi-linear* if it is a finite union of linear sets, and we call a unary language L semi-linear if the corresponding set S_L is semi-linear.

Over a unary alphabet, semi-linear languages are exactly the languages defined by *Presburger arithmetic* (Ginsburg and Spanier [14]), core spanners (Freydenberger and Holldack [10]) or generalized core spanners (Peterfreund, ten Cate, Fagin, and Kimelfeld [26]). By Freydenberger and Peterfreund [12], this means that FC (over unary alphabets) defines exactly the semi-linear languages. As 2^n grows faster than any linear function, no finite union of linear sets can define $\{2^n\}$. Hence, $L_{\text{pow}} := \{a^{2^n} \mid n \in \mathbb{N}\}$ is not semi-linear and, therefore, is not expressible in FC. From this, along with Lemma 3.4, we almost directly infer the following:

LEMMA 3.5. *Let $a \in \Sigma$. For every $k \in \mathbb{N}$, there exists $p, q \in \mathbb{N}$ such that $a^p \equiv_k a^q$ where $p \neq q$.*

Lemma 3.5 is one of our main building blocks for FC inexpressibility. That is, we shall utilize Duplicator's winning strategy for the k -round game over structures that represent a^p and a^q in order to construct winning strategies for more general structures.

A binary relation $R \subseteq (\Sigma^*)^2$ is a *congruence relation* if $a R b$ and $c R d$ implies $(a \cdot c) R (b \cdot d)$. When considering the more common FO[<] over words (where words are encoded as a linear order with symbol predicates), we have that $\equiv_k$ is a congruence relation (see Section 3.2 of [30]). Unfortunately, this does not hold for FC.

PROPOSITION 3.6. *There exists $u, u', v, v' \in \Sigma^*$ where for any $k \geq 5$, we have that $u \equiv_k v$ and $u' \equiv_k v'$, but $u \cdot u' \not\equiv_k v \cdot v'$.*

PROOF. For every $k \in \mathbb{N}$, there exists $p, q \in \mathbb{N}$ where $p \neq q$ and $a^p \equiv_k a^q$, see Lemma 3.5. Trivially, $b \cdot a^p \equiv_k b \cdot a^p$. But $a^p \cdot b \cdot a^p \not\equiv_k a^q \cdot b \cdot a^p$. To show that this does indeed hold, we give an FC-formula that accepts words of the form $v \cdot b \cdot v$ for $v \in \Sigma^*$, and thus can distinguish between $a^p \cdot b \cdot a^p$ and $a^q \cdot b \cdot a^p$ where $p \neq q$. Let

$$\varphi := \exists x, y, z: \left((y \doteq x \cdot z) \wedge (z \doteq b \cdot x) \wedge \neg \exists z_1, z_2: (((z_1 \doteq z_2 \cdot y) \vee (z_1 \doteq y \cdot z_2)) \wedge \neg (z_2 \doteq \varepsilon)) \right).$$

For any τ_Σ -structure $\mathfrak{A}_w$ that represents some word $w \in \Sigma^*$, the subformula

$$\psi(y) := \neg \exists z_1, z_2: (((z_1 \doteq z_2 \cdot y) \vee (z_1 \doteq y \cdot z_2)) \wedge \neg (z_2 \doteq \varepsilon))$$

states that $\sigma(y) = w$ for any substitution σ where $(\mathfrak{A}_w, \sigma) \models \psi$. Thus, for all $w \in \mathcal{L}(\varphi)$, we have that $w = vbv$ for some $v \in \Sigma^*$. Consequently, $a^p \cdot b \cdot a^p \in \mathcal{L}(\varphi)$ and $a^q \cdot b \cdot a^p \notin \mathcal{L}(\varphi)$ if $p \neq q$. Therefore, $a^p \cdot b \cdot a^p \not\equiv_k a^q \cdot b \cdot a^p$ for any $k \geq \text{qr}(\varphi)$. Since $\text{qr}(\varphi) = 5$, the stated lemma holds. $\square$

Proposition 3.6 rules out what would be a convenient technique for strategy composition. Despite this, in Section 4.1 we consider a special case where $\equiv$ can be used as if it were a congruence relation.

4 FC INEXPRESSIBILITY

Our next goal is using Ehrenfeucht-Fraïssé games in order to yield inexpressibility results for FC. To do so, we heavily utilize *strategy composition*. That is, we “bootstrap” known winning strategies for Duplicator – gathered from inexpressibility results such as Lemma 3.5 – and construct more general winning strategies for Duplicator. The main technical contribution of this section is the *Fooling Lemma* for FC that is derived from the *Pseudo-Congruence Lemma* and the *Primitive Power Lemma*. Section 4.1 contains the Pseudo-Congruence Lemma (Lemma 4.4), and Section 4.2 contains

the Primitive Power Lemma (Lemma 4.9). Both of these sections also contain many useful lemmas on the way to proving the respective result. Before looking at these lemmas, we first demonstrate the capabilities of FC by giving a language that (somewhat surprisingly) is expressible in FC.

For every $n \in \mathbb{N}$, we define $F_n \in \{a, b\}^*$ recursively as follows: $F_0 := a$, $F_1 := ab$, and $F_i := F_{i-1} \cdot F_{i-2}$ for all $i \geq 2$. The *Fibonacci word* is the limit F_ω (see [23] for details on F_ω and its properties). We show that the language L_{fib} containing those words $cF_0cF_1c \cdots cF_nc$ for $c \in \Sigma$ and for $n > 2$ is expressible in FC.

PROPOSITION 4.1. $L_{\text{fib}} \in \mathcal{L}(\text{FC})$.

PROOF. Let $\Sigma := \{a, b, c\}$. For this proof, we use arbitrary concatenation as a shorthand for binary concatenation. For example, we allow $(x \doteq y_1y_2y_3)$ as a shorthand of $(x \doteq y_1z_1) \wedge (z_1 \doteq y_2y_3)$.

First, consider

$$\varphi_w(x) := \neg \exists z_1, z_2 : \left(\neg(z_1 \doteq \varepsilon) \wedge ((z_2 \doteq z_1 \cdot x) \vee (z_2 \doteq x \cdot z_1)) \right).$$

This states that for any interpretation $(\mathfrak{A}_w, \sigma)$ where $\mathfrak{A}_w$ represents $w \in \Sigma^*$, and $(\mathfrak{A}_w, \sigma) \models \varphi_w$, it holds that $\sigma(x) = w$.

Now consider:

$$\varphi_{\text{struc}} := \exists x_1, u : (\varphi_w(u) \wedge (u \doteq \text{cacabc}x_1c) \wedge \neg \exists x_2 : (x_2 \doteq cc)).$$

The formula φ_{struc} states that any $w \in \mathcal{L}(\varphi_{\text{struc}})$ is of the form $\text{cacabc} \cdot (\{a, b\}^+ \cdot c)^+$.

Next, let

$$\varphi_{\text{fib}} := \varphi_{\text{struc}} \wedge \forall x, y_1, y_2, y_3 : \left((x \doteq cy_1cy_2cy_3c) \rightarrow \right. \\ \left. (\varphi_c(y_1) \vee \varphi_c(y_2) \vee \varphi_c(y_3) \vee (y_3 \doteq y_2 \cdot y_1)) \right),$$

where $\varphi_c(x) := \exists y, z : (x \doteq y \cdot c \cdot z)$, and $\rightarrow$ is logical implication. The formula φ_{fib} accepts those words in $\mathcal{L}(\varphi_{\text{struc}})$ where for any factor of the form $cy_1cy_2cy_3c$ for $y_1, y_2, y_3 \in \Sigma^*$, either

- for some $i \in [3]$, the word y_i contains a c , or
- $y_3 = y_2 \cdot y_1$.

Thus, for all factors of the form $cy_1cy_2cy_3c$ where $y_1, y_2, y_3 \in \{a, b\}^+$, we have that $y_3 = y_2 \cdot y_1$. Due to φ_{struc} , any $w \in \mathcal{L}(\varphi_{\text{fib}})$ as the prefix cacabc , and therefore $\mathcal{L}(\varphi_{\text{fib}}) = L_{\text{fib}}$. $\square$

While the definition suggests that expressing L_{fib} requires a logic that allows recursion, we can use the universal quantifier to simulate recursion in certain cases. As a curious aside, this also shows that FC does not have a pumping lemma in the sense that for sufficiently long words, some non-empty factor can be repeated an arbitrary number of times without falling out of the language. This is due to the fact that F_ω does not contain any factor of the form u^4 with $u \neq \varepsilon$, see Karhumäki [17].

4.1 The Pseudo-Congruence Lemma

In this section, we first give some necessary lemmas, we then give the Pseudo-Congruence Lemma along with a proof idea. To conclude this section, we look at some consequences.

First, we consider the case where Spoiler has chosen a factor that is so short (with respect to the number of remaining rounds) that Duplicator must respond with the identical factor or lose.

LEMMA 4.2. Let $\mathfrak{A}_w$ and $\mathfrak{B}_v$ be τ_Σ -structures that represent $w \in \Sigma^*$ and $v \in \Sigma^*$, where $\mathfrak{A}_w \equiv_k \mathfrak{B}_v$. Let $\vec{a} = (a_1, a_2, \dots, a_{k+|\Sigma|+1})$ and $\vec{b} = (b_1, b_2, \dots, b_{k+|\Sigma|+1})$ be the tuples that result from a k -round game over $\mathfrak{A}_w$ and $\mathfrak{B}_v$ where Duplicator plays their winning strategy. If $r + |a_r| - 1 < k$ or $r + |b_r| - 1 < k$ for some $r \in [k]$, then $b_r = a_r$.

Next, we show that for rounds r with $r \leq k - 2$, if Spoiler picks a prefix (or a suffix), then Duplicator must respond with a prefix (or suffix, respectively); assuming they play a winning strategy.

LEMMA 4.3. *Let $\mathfrak{A}_w$ and $\mathfrak{B}_v$ be two τ_Σ -structures that represent $w \in \Sigma^*$ and $v \in \Sigma^*$, where $\mathfrak{A}_w \equiv_k \mathfrak{B}_v$. Let $\vec{a} = (a_1, a_2, \dots, a_{k+|\Sigma|+1})$ and $\vec{b} = (b_1, b_2, \dots, b_{k+|\Sigma|+1})$ be the tuples that result from a k -round game over $\mathfrak{A}_w$ and $\mathfrak{B}_v$ where Duplicator plays their winning strategy. For any $r \leq k - 2$, we have that a_r is a suffix (or prefix) of w if and only if b_r is a suffix (or prefix respectively) of v .*

While we mainly use Lemma 4.2 and Lemma 4.3 to help prove the Pseudo-Congruence Lemma, they also provide some insights into necessary conditions of Duplicator's strategy.

In some of the subsequent proofs, we use strategy compositions that require quite technical proofs of correctness. To avoid handwaving, we shall sometimes define Duplicator's strategy using what we call *look-up games*. If Spoiler and Duplicator are playing a k -round game $\mathcal{G}$ over $\mathfrak{A}_w$ and $\mathfrak{B}_v$, a *look-up game* is a k' -round auxiliary game $\mathcal{G}'$ over two (potentially different) structures $\mathfrak{C}$ and $\mathfrak{D}$, where $\mathfrak{C} \equiv_{k'} \mathfrak{D}$. The idea is that in the i -th round in $\mathcal{G}$, each move by Spoiler corresponds to a move by Spoiler in $\mathcal{G}'$. Then, Duplicator "looks up" what their response would be in $\mathcal{G}'$ to form their response to Spoiler in $\mathcal{G}$. Generalizing this idea, Duplicator can use multiple look-up games $\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_n$ to form their response in $\mathcal{G}$. In other words, if Duplicator has winning strategies in the games $\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_n$, then Duplicator has a winning strategy in $\mathcal{G}$.

While $\equiv_k$ is not a congruence relation in general (see Proposition 3.6), we identify a special case where we can use $\equiv$ as if it were a congruence relation.

LEMMA 4.4 (PSEUDO-CONGRUENCE LEMMA). *Let $w_1, w_2, v_1, v_2 \in \Sigma^*$ where $\text{Facs}(w_1) \cap \text{Facs}(w_2) = \text{Facs}(v_1) \cap \text{Facs}(v_2)$, and let $r := \max\{|u| \in \mathbb{N} \mid u \in \text{Facs}(w_1) \cap \text{Facs}(w_2)\}$. If $w_1 \equiv_{k+r+2} v_1$ and $w_2 \equiv_{k+r+2} v_2$ for some $k \in \mathbb{N}_+$, then $w_1 \cdot w_2 \equiv_k v_1 \cdot v_2$.*

PROOF SKETCH. We consider two look-up games $\mathcal{G}_1$ and $\mathcal{G}_2$. For $\iota \in \{1, 2\}$, the game $\mathcal{G}_\iota$ is a $k + r + 2$ round game over w_ι and v_ι . We assume Duplicator plays $\mathcal{G}_1$ and $\mathcal{G}_2$ using their winning strategy. We use these look-up games to give Duplicator's strategy for the k round game $\mathcal{G}$ over $w := w_1 \cdot w_2$ and $v := v_1 \cdot v_2$.

For each round $p \in [k]$ of $\mathcal{G}$, we determine Spoiler's choice in $\mathcal{G}_1$ and $\mathcal{G}_2$ from Spoiler's choice in $\mathcal{G}$. If Spoiler chooses some $u \in \text{Facs}(w_\iota) \cup \text{Facs}(v_\iota)$, where $\iota \in \{1, 2\}$, then we let Spoiler's choice in round p of $\mathcal{G}_\iota$ be u . If Spoiler chooses some $u \in \text{Facs}(w) \setminus (\text{Facs}(w_1) \cup \text{Facs}(w_2))$, then we consider $u_1 \in \text{Facs}(w_1)$ and $u_2 \in \text{Facs}(w_2)$ such that $u = u_1 \cdot u_2$. Then, we let Spoiler choose u_1 in round p of $\mathcal{G}_1$ and u_2 in round p of $\mathcal{G}_2$; see Fig. 1. We do the analogous action if Spoiler chooses $u \in \text{Facs}(v) \setminus (\text{Facs}(v_1) \cup \text{Facs}(v_2))$. If, for example, Spoiler chooses $u \in \text{Facs}(w_1) \setminus \text{Facs}(w_2)$, then we assume that Spoiler "skips" round p of $\mathcal{G}_2$. Analogously, Spoiler skips round p of $\mathcal{G}_1$ if Spoiler chose $u \in \text{Facs}(w_2) \setminus \text{Facs}(w_1)$ in round p of $\mathcal{G}$.

To sum up Duplicator's strategy for $\mathcal{G}$ informally: If Spoiler chooses an element from either $\text{Facs}(w_1)$ or $\text{Facs}(v_1)$, then Duplicator responds with their winning strategy for $\mathcal{G}_1$. Likewise, if Spoiler chooses from either $\text{Facs}(w_2)$ or $\text{Facs}(v_2)$, then Duplicator responds with their winning strategy for $\mathcal{G}_2$. Finally, if Spoiler chooses $u \in \text{Facs}(w) \setminus (\text{Facs}(w_1) \cup \text{Facs}(w_2))$ or $u \in \text{Facs}(v) \setminus (\text{Facs}(v_1) \cup \text{Facs}(v_2))$, then we split $u = u_1 \cdot u_2$ where $(u_1, u_2) \in \text{Facs}(w_1) \times \text{Facs}(w_2)$, or $(u_1, u_2) \in \text{Facs}(v_1) \times \text{Facs}(v_2)$, and Duplicator responds with the concatenation of their winning strategy when Spoiler chooses u_1 in $\mathcal{G}_1$ and their winning strategy when Spoiler chooses u_2 in $\mathcal{G}_2$.

The proof that this composition of strategies is a winning strategy for Duplicator is somewhat tedious. However, in this sketch, we informally explain how this strategy is indeed a strategy for Duplicator over w and v . That is, Duplicator's strategy is well-defined and Duplicator always chooses a factor of w or v .

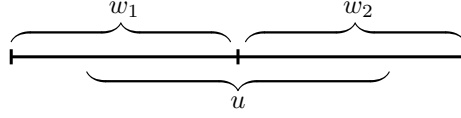


Fig. 1. Illustration of factors in $\text{Facs}(w) \setminus (\text{Facs}(w_1) \cup \text{Facs}(w_2))$.

The extra $r + 2$ rounds for $\mathcal{G}_1$ and $\mathcal{G}_2$ are to ensure that Duplicator makes certain choices. If in $\mathcal{G}$ Spoiler chooses some $u \in \text{Facs}(w_1) \cap \text{Facs}(w_2)$ for some round $p \leq k$, then $|u| \leq r$ holds (from the lemma statement). Therefore, in both $\mathcal{G}_1$ and $\mathcal{G}_2$, Duplicator responds with u , see Lemma 4.2. The same holds when Spoiler chooses some $u \in \text{Facs}(v_1) \cap \text{Facs}(v_2)$. Therefore, Duplicator's choice is well defined for the case where Spoiler chooses some $u \in \text{Facs}(w_1) \cap \text{Facs}(w_2)$ due to the fact that Duplicator's responses in $\mathcal{G}_1$ and $\mathcal{G}_2$ coincide. The same holds when Spoiler chooses some factor in $\text{Facs}(v_1) \cap \text{Facs}(v_2)$.

Let us consider the case where for round $p \leq k$ in $\mathcal{G}$, Spoiler chooses some $u \in \text{Facs}(w) \setminus (\text{Facs}(w_1) \cup \text{Facs}(w_2))$. Observing Fig. 1, we can split u into the suffix u_1 of w_1 and the prefix u_2 of w_2 . Thus, due to Lemma 4.3, in $\mathcal{G}_1$, Duplicator responds to u_1 with a suffix u'_1 of v_1 . Likewise, in $\mathcal{G}_2$, Duplicator responds to u_2 with a prefix u'_2 of v_2 . Consequently, $u'_1 \cdot u'_2$ is indeed a factor of v . The analogous reasoning holds if Spoiler chooses some $u \in \text{Facs}(v) \setminus (\text{Facs}(v_1) \cup \text{Facs}(v_2))$.

We have given a description of Duplicator's strategy, and informally sketched a proof that this strategy is well-defined. $\square$

Freydenberger and Peterfreund [12] showed that the language $\{a^n \cdot b^n \mid n \in \mathbb{N}\}$ is not expressible in FC. Their proof uses an alternative logic $\text{FO}[\text{EQ}]$ with equivalent expressive power, for which the Feferman-Vaught theorem can be invoked. As this approach heavily relies on the restricted nature of words $a^n \cdot b^m$, it seems to be very difficult to generalize. Furthermore, even for $\text{FO}[\text{EQ}]$, the Feferman-Vaught theorem needs some additional reasoning. Compare this to the following alternative proof.

Example 4.5. From Lemma 3.5, we know that for every $k \in \mathbb{N}$, there exists $p, q \in \mathbb{N}$ where $a^p \equiv_{k+2} a^q$ and, without loss of generality, $p < q$. We also have that $b^p \equiv_{k+2} b^p$ for all $k \in \mathbb{N}$. Furthermore, $\text{Facs}(a^n) \cap \text{Facs}(b^m) = \{\varepsilon\}$ for any $m, n \in \mathbb{N}$. We can now invoke Lemma 4.4, with $r = 0$, to state that for any $k \in \mathbb{N}$, there exists $p, q \in \mathbb{N}$ such that $a^q \cdot b^p \equiv_k a^p \cdot b^p$ and $p < q$. Consequently, $\{a^i \cdot b^j \mid 0 \leq i \leq j\} \notin \mathcal{L}(\text{FC})$ and $\{a^n \cdot b^n \mid n \in \mathbb{N}\} \notin \mathcal{L}(\text{FC})$.

In addition to giving an alternative proof of an existing result, we can also use Lemma 4.4 to prove new inexpressibility results.

PROPOSITION 4.6. $\{a^n \cdot (ba)^n \mid n \in \mathbb{N}\} \notin \mathcal{L}(\text{FC})$.

PROOF. To prove this result, we show that for every $k \in \mathbb{N}$, there exists $p, q \in \mathbb{N}$ where $p < q$ and $a^q(ba)^q \equiv_k a^p(ba)^q$.

From Lemma 3.5, for every $k \in \mathbb{N}$, there exists $p, q \in \mathbb{N}$ such that $a^q \equiv_{k+3} a^p$ where $q \neq p$. Without loss of generality, assume that $p < q$. Trivially, $(ba)^q \equiv_{k+3} (ba)^q$ for every $k \in \mathbb{N}$. Since, $\text{Facs}(a^m) \cap \text{Facs}((ba)^n) = \{\varepsilon, a\}$ for any $m, n \in \mathbb{N}$, we can invoke Lemma 4.4 with $r = 1$. It therefore follows that for every $k \in \mathbb{N}$, there exists $p, q \in \mathbb{N}$ where $p \neq q$ and $a^q(ba)^q \equiv_k a^p(ba)^q$. From Lemma 3.4 we infer that $\{a^n \cdot (ba)^n \mid n \in \mathbb{N}\} \notin \mathcal{L}(\text{FC})$. $\square$

Observing Proposition 4.6, we see that the Pseudo-Congruence Lemma by itself is a useful tool for inexpressibility.

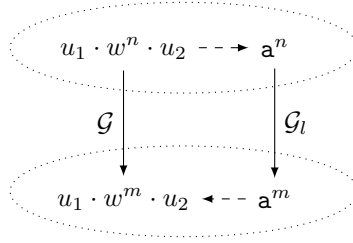


Fig. 2. Duplicator's strategy when Spoiler chooses some u where $\exp_w(u) \geq 1$. The top oval illustrates Spoiler's choices in $\mathcal{G}$ and $\mathcal{G}_l$, and the bottom oval illustrates Duplicator's responses in $\mathcal{G}$ and $\mathcal{G}_l$.

4.2 The Primitive Power Lemma

In this section, we look at the Primitive Power Lemma. Before giving this result, we first consider a lemma for words that are primitive (that is, not a power of a shorter word).

For any $w \in \Sigma^+$, let $\exp_w: \Sigma^+ \rightarrow \mathbb{N}$ be a function that maps $u \in \Sigma^+$ to $m \in \mathbb{N}$ if $w^m \sqsubseteq u$ and there does not exist $m' > m$ such that $w^{m'} \sqsubseteq u$. Less formally, $\exp_w(u)$ is the maximum value $m \in \mathbb{N}$ such that w^m is a factor of u .

Example 4.7. Consider $u := \text{aaaabaabaab}$. Then, $\exp_a(u) = 4$, and $\exp_{aab}(u) = 3$.

The following is due to some basic results on primitive words.

LEMMA 4.8. *Let $w \in \Sigma^+$ be a primitive word and let $m \in \mathbb{N}$. If $u \sqsubseteq w^m$ where $\exp_w(u) > 0$, then there is a unique proper suffix $u_1 \in \Sigma^*$ of w and a unique proper prefix $u_2 \in \Sigma^*$ of w , such that $u = u_1 \cdot w^{\exp_w(u)} \cdot u_2$.*

While perhaps not particularly interesting by itself, Lemma 4.8 shall be an important part of the proof of the Primitive Power Lemma – a result that (in a sense) generalizes Lemma 3.5.

LEMMA 4.9 (PRIMITIVE POWER LEMMA). *If $a^p \equiv_{k+3} a^q$ for $a \in \Sigma$ and $p, q, k \in \mathbb{N}_+$, then $w^p \equiv_k w^q$ for any primitive word $w \in \Sigma^+$.*

PROOF SKETCH. To prove this lemma, we give a winning strategy for Duplicator for the k -round game $\mathcal{G}$ over w^p and w^q . This strategy is defined based on the $k+3$ -round look-up game $\mathcal{G}_l$ over a^p and a^q , where $a^p \equiv_{k+3} a^q$.

For each round $p \leq k$ in $\mathcal{G}$, if Spoiler chooses u where $\exp_w(u) = n$, then we let Spoiler choose a^n in round p of $\mathcal{G}_l$ in the corresponding word. Duplicator's response in $\mathcal{G}_l$ is some a^m . Then, we form Duplicator's response to Spoiler in $\mathcal{G}$ as follows:

- If in $\mathcal{G}_l$, Duplicator responds with ε , then let Duplicator's response in $\mathcal{G}$ be the same factor that Spoiler chose. This is because it can be shown that Spoiler chose some u where $\exp_w(u) = 0$.
- If in $\mathcal{G}_l$, Duplicator responds with a^m where $m \in \mathbb{N}_+$, then we can show that in $\mathcal{G}$, Spoiler chose $u = u_1 \cdot w^n \cdot u_2$ where u_1 and u_2 are a unique suffix and prefix of w (see Lemma 4.8). We define Duplicator's response in $\mathcal{G}$ as $u_1 \cdot w^m \cdot u_2$. See Fig. 2 for an informal illustration of Duplicator's strategy.

In the full proof, we show that this strategy is indeed a winning strategy for Duplicator by utilizing several lemmas regarding primitive words (such as Lemma 4.8). $\square$

For any $w \in \Sigma^+$, the *primitive root* of w is the unique primitive word $z \in \Sigma^+$ such that $w = z^k$ for some $k \geq 1$. Therefore, almost immediately from Lemma 4.9, we are able to show:

PROPOSITION 4.10. *For any $w \in \Sigma^+$ and any $k \in \mathbb{N}$, there exists $p \in \mathbb{N}_+$ and $v \in \Sigma^*$ such that $w^p \equiv_k v$ where $w^p \neq v$.*

PROOF. Let $w \in \Sigma^+$ and let $z \in \Sigma^+$ be the primitive root of w . That is, $w = z^i$ for some $i \in \mathbb{N}_+$ and some primitive word $z \in \Sigma^+$. It follows that for any $p \in \mathbb{N}_+$, we have that $w^p = z^{pi}$. Therefore, if for any primitive word $z \in \Sigma^+$, and any $k \in \mathbb{N}$, there exists $p \in \mathbb{N}_+$ and $v \in \Sigma^*$ such that $v \neq z^{pi}$, where $z^{pi} \equiv_k v$, then we have proven the claim. To that end, we observe that the set $\{i \cdot 2^n \mid n \in \mathbb{N}_+\}$ is not semi-linear (recall Section 3, using the same reasoning as for L_{pow}) and, hence, is not definable in FC. Consequently, from Lemma 4.9, for any word $w \in \Sigma^+$ and any $k \in \mathbb{N}_+$, there exists $p \in \mathbb{N}_+$ and $v \in \Sigma^*$ such that $w^p \equiv_k v$ where $w^p \neq v$. $\square$

As an immediate consequence of Lemma 4.9, it follows that for any primitive word $w \in \Sigma^+$, the language w^{2^n} is not expressible in FC. While this may seem somewhat restrictive, due to the fact that this result holds for primitive words, any non-primitive word is a repetition of primitive words, hence Proposition 4.10.

4.3 The Fooling Lemma for FC

The Fooling Lemma for FC combines the Pseudo-Congruence Lemma with the Primitive Power Lemma. In order to give a convenient formulation of this result, we first look at *co-primitive* words: For two words $w, v \in \Sigma^+$, we say that w and v are *conjugate* if there exists $x, y \in \Sigma^*$ such that $w = x \cdot y$ and $v = y \cdot x$. If w and v are not conjugate and both w and v are primitive words, then we say that w and v are *co-primitive*.

Example 4.11. It is clear that $u := aabba$ and $v := aaabb$ are both primitive words. However, u and v are not co-primitive. This can easily be observed by considering $x := aabb$ and $y := a$. Then, $u = xy$ and $v = yx$. Now consider $u' := aba$ and $v' := bba$. Again, both u' and v' are primitive. Furthermore, from the fact that $|u'|_a \neq |v'|_a$, we can also see that u' and v' are co-primitive.

For any two words $w, v \in \Sigma^*$, let $w^\omega = w \cdot w \cdots$ and $v^\omega = v \cdot v \cdots$ be the one-sided infinite word consisting of continuous repetitions of w and v respectively.

We now state the *periodicity lemma* [7]. For our purposes, we use the formulation of this result given in [15]. Further information regarding the periodicity lemma can be found in Section 6.3, Chapter 6 of [27].

LEMMA 4.12 (PERIODICITY LEMMA). *Let w and v be primitive words. If w^ω and v^ω have a common factor of length at least $|w| + |v| - 1$, then w and v are conjugate.*

The periodicity lemma and some further reasoning gives us:

LEMMA 4.13. *For any primitive words $w, v \in \Sigma^*$, the following are equivalent:*

- (1) w and v are co-primitive,
- (2) there exists $n_0, m_0 \in \mathbb{N}$ such that $\text{Facs}(w^{n_0}) \cap \text{Facs}(v^{m_0}) = \text{Facs}(w^n) \cap \text{Facs}(v^m)$ for all $n > n_0$ and all $m > m_0$, and
- (3) there exists $r \in \mathbb{N}$ such that $r \geq \max\{|w| \in \mathbb{N} \mid w \in \text{Facs}(u^n) \cap \text{Facs}(v^m)\}$ for all $n, m \in \mathbb{N}$.

We are almost ready to give the Fooling Lemma for FC. Before doing so, the Fooling Lemma is simply a possible way of combining the results we have given so far, and there could be a more general inexpressibility lemma for FC just using the results from this paper. However, for our purposes, Lemma 4.14 is enough.

LEMMA 4.14 (FOOLING LEMMA). *Let $w_1, w_2, w_3 \in \Sigma^*$, let $u, v \in \Sigma^+$ be two co-primitive words, let $\varphi \in \text{FC}$, and let $f: \mathbb{N} \rightarrow \mathbb{N}$ be an injective function. If $w_1 \cdot u^p \cdot w_2 \cdot v^{f(p)} \cdot w_3 \in \mathcal{L}(\varphi)$ for all $p \in \mathbb{N}$, then $w_1 \cdot u^s \cdot w_2 \cdot v^t \cdot w_3 \in \mathcal{L}(\varphi)$ for some $s, t \in \mathbb{N}$, where $f(s) \neq t$.*

PROOF SKETCH. We use the Primitive Power Lemma along with the Pseudo-Congruence Lemma to determine that for every $k \in \mathbb{N}$ there exists $p, q \in \mathbb{N}$ such that $w_1 \cdot u^p \cdot w_2 \equiv_k w_1 \cdot u^q \cdot w_2$ where $p \neq q$. Then, we again invoke the Pseudo-Congruence Lemma (using Lemma 4.13 to ensure that the required prerequisites hold) to conclude that for every $k \in \mathbb{N}$, there exists $p, q \in \mathbb{N}$ where $p \neq q$ and $w_1 \cdot u^p \cdot w_2 \cdot v^{f(p)} \cdot w_3 \equiv_k w_1 \cdot u^q \cdot w_2 \cdot v^{f(q)} \cdot w_3$. Since f is injective, $f(p) \neq f(q)$ if $p \neq q$. $\square$

Immediately from Lemma 4.14, we infer the following:

PROPOSITION 4.15. *Let $w_1, w_2, w_3 \in \Sigma^*$, and let $u, v \in \Sigma^+$ be two co-primitive words. Then, for any injective function $f: \mathbb{N} \rightarrow \mathbb{N}$, the language $\{w_1 \cdot u^p \cdot w_2 \cdot v^{f(p)} \cdot w_3 \mid p \in \mathbb{N}\}$ is not an FC-language.*

So far, we have developed several inexpressibility techniques. Next, we look at using these tools to show that the following languages are not FC languages.

LEMMA 4.16. *The following languages are not expressible in FC:*

- $L_1 := \{a^n \cdot (ba)^n \mid n \in \mathbb{N}\}$,
- $L_2 := \{a^i \cdot (ba)^j \mid 1 \leq i \leq j\}$,
- $L_3 := \{b^n \cdot a^m \cdot b^{n+m} \mid m, n \in \mathbb{N}\}$,
- $L_4 := \{b^n \cdot a^m \cdot b^{n-m} \mid m, n \in \mathbb{N}\}$,
- $L_5 := \{(abaabb)^m \cdot (bbaaba)^m \mid m \in \mathbb{N}\}$, and
- $L_6 := \{a^n \cdot b^n \cdot (ab)^n \mid n \in \mathbb{N}\}$.

PROOF. We show that each language L_i , for $i \in \{1, \dots, 6\}$ cannot be expressed in FC.

Regarding L_1 and L_2 : The fact that $L_1, L_2 \notin \mathcal{L}(\text{FC})$ follows from Proposition 4.6 and Lemma 3.4

Regarding L_3 : Consider the case where $n = 0$. We therefore get the language $\{a^m \cdot b^m \mid m \in \mathbb{N}\}$. From Example 4.5, we know that for every $k \in \mathbb{N}$, there exists $p, q \in \mathbb{N}$ such that $q \neq p$ and $a^q \cdot b^p \equiv_k a^p \cdot b^p$. Thus, invoking Lemma 3.4, we conclude that $L_3 \notin \mathcal{L}(\text{FC})$.

Regarding L_4 : Working towards a contradiction, assume that there exists $\varphi \in \text{FC}$ such that $\mathcal{L}(\varphi) = L_4$. Consider the case where $n = 1$. We then get words of the form $b \cdot a^m \cdot b^m$. Let $w_1 = b$ and $w_2 = w_3 = \varepsilon$. Let $u = a$, let $v = b$, and let $f: \mathbb{N} \rightarrow \mathbb{N}$ be the identity function. Then, invoking the Fooling Lemma, we get that $b \cdot a^s \cdot b^t \in \mathcal{L}(\varphi)$ where $s \neq t$. Consequently, $L_4 \notin \mathcal{L}(\text{FC})$.

Regarding L_5 : The words $abaabb$ and $bbaaba$ are both co-primitive. Thus, there exists some $r \in \mathbb{N}$ where $r \geq \max\{|u| \in \mathbb{N} \mid u \in \text{Facs}((abaabb)^m) \cap \text{Facs}((bbaaba)^n)\}$ for all $m, n \geq 1$. Therefore, immediately from Proposition 4.15 we conclude that $L_5 \notin \mathcal{L}(\text{FC})$.

Regarding L_6 : We know that for every $k \in \mathbb{N}$, there exists $p, q \in \mathbb{N}$ such that $a^p \cdot b^p \equiv_{k+4} a^q \cdot a^p$ and $p \neq q$. Furthermore, $(ab)^p \equiv_{k+4} (ab)^p$ for all $k \in \mathbb{N}$.

Since for any $i, j, l \in \mathbb{N}_+$, it holds that $\text{Facs}(a^i \cdot b^j) \cap \text{Facs}((ab)^l) = \{\varepsilon, a, b, ab\}$, we can invoke Lemma 4.4 with $r = 2$. Thus, for every $k \in \mathbb{N}$, there exists $p, q \in \mathbb{N}_+$ with $p \neq q$, and $a^q \cdot b^p \cdot (ab)^p \equiv_k a^p \cdot b^p \cdot (ab)^p$. Therefore, $L_6 \notin \mathcal{L}(\text{FC})$. $\square$

The authors note that it seems at least possible to show that L_3 and L_4 in Lemma 4.16 are not FC languages by using the Feferman-Vaught Theorem and the logic $\text{FO}[\text{EQ}]$ (see [12] for more details). However, in order to do so, a sizeable amount of work would be needed. In comparison, using the techniques developed in this work, it is rather straightforward to show $L_3, L_4 \notin \mathcal{L}(\text{FC})$.

While Lemma 4.16 seems somewhat “language theoretic”, in Theorem 5.6 we shall use Lemma 4.16 to show that several relations are not definable in $\text{FC}[\text{REG}]$.

5 GENERALIZED CORE SPANNER INEXPRESSIBILITY

Freydenberger and Peterfreund [12] proved that FC extended with so-called *regular constraints* captures the expressive power of generalized core spanners. That is, a word relation $R \subseteq (\Sigma^*)^k$ is definable in $\text{FC}[\text{REG}]$ if and only if R is *selectable* by generalized core spanners, see Theorem 5.9

of [12]. We do not give the definition of generalized core spanners here, since we are able to simply use $\text{FC}[\text{REG}]$ instead. See Section 5 of [6] for the formal definitions of generalized core spanners¹.

Definition 5.1. A *regular constraint* is an atomic formula $(x \in \gamma)$ where $x \in \Sigma$ and γ is a regular expression. Let $\text{FC}[\text{REG}]$ be the logic that extends FC with regular constraints. The semantics are as follows: For a regular constraint $(x \in \gamma)$, and an FC interpretation $\mathcal{I} := (\mathfrak{A}_w, \sigma)$, where $\mathfrak{A}_w$ is a τ_Σ -structure that represents $w \in \Sigma^*$, we write $\mathcal{I} \models (x \in \gamma)$ if $\sigma(x) \sqsubseteq w$, and $\sigma(x) \in \mathcal{L}(\gamma)$.

Unfortunately, the introduction of regular constraints also introduces some difficulties when dealing with the quantifier rank of a formula. Whilst one could easily adapt the definition of quantifier rank to $\text{FC}[\text{REG}]$, we run into the issue that there are infinite $\text{FC}[\text{REG}]$ -formulas of quantifier rank one. Consider $\exists x: (x \in \gamma)$, along with the infinite number of regular languages. This would mean that we cannot use Theorem 3.3 for $\text{FC}[\text{REG}]$, which requires (up to logical equivalence) a finite number of formulas of quantifier rank k for any $k \in \mathbb{N}$ (see, for example, Chapter 3 of [21] for more details).

We now consider a way to handle the regular constraints in order to derive inexpressibility results for generalized core spanners.

Definition 5.2. A language $L \subseteq \Sigma^*$ is *bounded* if it is a subset of a language $w_1^* \cdot w_2^* \cdots w_n^*$ where $n \in \mathbb{N}$ and $w_i \in \Sigma^*$ for all $i \in [n]$.

A *bounded regular language* is simply a language that is both bounded and regular. From previous literature (see [9, 12] for example), we know that every bounded regular language can be expressed in FC. Thus, using a very similar proof idea to Theorem 6.2 from [9] or Lemma 5.5 from [12], we are able to show the following:

LEMMA 5.3. *If $L \subseteq \Sigma^*$ is a Boolean combination of bounded languages, then $L \in \mathcal{L}(\text{FC})$ if and only if $L \in \mathcal{L}(\text{FC}[\text{REG}])$.*

The proof of Lemma 5.3 is based on the idea of commuting words (see Proposition 1.3.2 of [22] for example). That is, if $xy = yx$ for some $x, y \in \Sigma^+$, then there is some $z \in \Sigma^*$ and $k_1, k_2 \in \mathbb{N}$ such that $x = z^{k_1}$ and $y = z^{k_2}$. This allows us to simulate $(w)^*$, for any $w \in \Sigma^*$, in FC.

Example 5.4. Consider the $\text{FC}[\text{REG}]$ -formula $\varphi := \exists x, y: ((x \doteq y \cdot y) \wedge (y \in (\text{abab})^+))$. We can re-write φ as:

$$\varphi' := \exists x, y, y', z: ((x \doteq y \cdot y) \wedge (y \doteq y' \cdot y') \wedge (y' \doteq z \cdot \text{ab}) \wedge (y' \doteq \text{ab} \cdot z)).$$

Since we have $y' \doteq z \cdot \text{ab}$ and $y' \doteq \text{ab} \cdot z$, it follows from [22] that there is some $k \in \mathbb{N}$ such that z is always mapped to some word of the form ab^k . Hence, φ' simulates the regular constraint $y' \in (\text{ab})^+$ which then simulates the regular constraint $y \in (\text{abab})^+$. Consequently, $\mathcal{L}(\varphi) = \mathcal{L}(\varphi')$.

From Lemma 5.5 in [12], we know that if every regular constraint in an $\text{FC}[\text{REG}]$ -formula is a so-called *simple regular expression*, then we can rewrite the $\text{FC}[\text{REG}]$ -formula into an FC-formula. A simple regular expression is a regular expression where the $*$ operator is only applied to a word $w \in \Sigma^*$, or to Σ . For example, $\Sigma^* \cdot (\text{aba})^*$ is a simple regular expression, whereas $(\text{ab} \cup \text{ba})^*$ is not. We can therefore replace every constraint that is a Boolean combination of simple regular expressions with an FC-formula. We do not go further into this, as Lemma 5.3 is all that we require.

Before looking at relations that are not definable by $\text{FC}[\text{REG}]$, we first define some concepts.

For $x, y \in \Sigma^*$, we say that x is a *scattered subword* of y (denoted by $x \sqsubseteq_{\text{scatt}} y$) if for some $n \geq 1$, there exists $x_1, x_2, \dots, x_n \in \Sigma^*$ and $y_0, y_1, \dots, y_n \in \Sigma^*$ such that $x = x_1 x_2 \cdots x_n$ and $y = y_0 x_1 y_1 x_2 y_2 \cdots x_n y_n$.

¹Note that in [6], what we call generalized core spanners is called core spanners with difference. The term “generalized core spanner” was coined in [26].

A similar concept to scattered subwords is the *shuffle product*. For $x, y \in \Sigma^*$, we define the shuffle product of x and y , denoted $x \sqcup y$, as the set of all words $x_1y_1x_2y_2 \cdots x_ny_n$ where $x = x_1 \cdots x_n$ and $y = y_1 \cdots y_n$ for $x_i, y_i \in \Sigma^*$ for all $i, j \in [n]$.

Example 5.5. Let $u := abba$ and $v := aa$. Then, $v \sqsubseteq_{\text{scatt}} u$ and $ababaa \in u \sqcup v$.

We are now ready to use Lemma 4.16 to show that several relations are not definable in $\text{FC}[\text{REG}]$, and therefore are also not selectable by generalized core spanners.

THEOREM 5.6. *The following are not definable in $\text{FC}[\text{REG}]$:*

- $\text{Num}_a := \{(x, y) \in (\Sigma^*)^2 \mid |x|_a = |y|_a\}$ for any $a \in \Sigma$,
- $\text{Add} := \{(x, y, z) \in (\Sigma^*)^3 \mid |z| = |x| + |y|\}$,
- $\text{Mult} := \{(x, y, z) \in (\Sigma^*)^3 \mid |z| = |x| \cdot |y|\}$,
- $\text{Scatt} := \{(x, y) \in (\Sigma^*)^2 \mid x \sqsubseteq_{\text{scatt}} y\}$,
- $\text{Perm} := \{(x, y) \in (\Sigma^*)^2 \mid x \text{ is a permutation of } y\}$,
- $\text{Rev} := \{(x, y) \in (\Sigma^*)^2 \mid x \text{ is the reverse of } y\}$,
- $\text{Shuff} := \{(x, y, z) \in (\Sigma^*)^3 \mid z \in x \sqcup y\}$,
- $\text{Morph}_h := \{(x, y) \in (\Sigma^*)^2 \mid y = h(x)\}$ for a given morphism $h: \Sigma^* \rightarrow \Sigma^*$.

A morphism is a function $h: \Sigma^* \rightarrow \Sigma^*$ with $h(xy) = h(x) \cdot h(y)$ for all $x, y \in \Sigma^*$.

PROOF. First, let $\varphi_w(\mathbf{u}) := \neg \exists z_1, z_2: \left(\neg(z_2 \doteq \epsilon) \wedge ((z_1 \doteq z_2 \cdot x) \vee (z_1 \doteq x \cdot z_2)) \right)$. For every interpretation $(\mathfrak{A}_w, \sigma)$, where $\mathfrak{A}_w$ is a τ_Σ -structure that represents $w \in \Sigma^*$, and where $(\mathfrak{A}_w, \sigma) \models \varphi_w$, we have that $\sigma(\mathbf{u}) = w$.

Note that for this proof, we use arbitrary concatenation as a shorthand. For example, $(x \doteq y_1 \cdot y_2 \cdot y_3)$ is shorthand for $(x \doteq y_1 \cdot z_1) \wedge (z_1 \doteq y_2 \cdot y_3)$ where z_1 is a new and unique variable. Also note that all the languages given in Lemma 4.16 are bounded languages, therefore $L_i \notin \mathcal{L}(\text{FC}[\text{REG}])$ for $i \in [6]$.

Assume that some relation R from $\text{Num}_a, \text{Add}, \text{Mult}, \text{Scatt}, \text{Perm}, \text{Rev}, \text{Shuff}$ is definable in $\text{FC}[\text{REG}]$, and let $\varphi_R \in \text{FC}[\text{REG}]$ define R . Now consider the following $\text{FC}[\text{REG}]$ -formulas:

$$\begin{aligned} \psi_1 &:= \exists \mathbf{u}, x, y: (\varphi_w(\mathbf{u}) \wedge (\mathbf{u} \doteq xy) \wedge (x \in a^*) \wedge (y \in (ba)^*) \wedge \varphi_{\text{Num}_a}(x, y)), \\ \psi_2 &:= \exists \mathbf{u}, x, y: (\varphi_w(\mathbf{u}) \wedge (\mathbf{u} \doteq xy) \wedge (x \in a^*) \wedge (y \in (ba)^*) \wedge \varphi_{\text{Scatt}}(x, y)), \\ \psi_3 &:= \exists \mathbf{u}, x, y, z: (\varphi_w(\mathbf{u}) \wedge (\mathbf{u} \doteq xyz) \wedge (x \in b^*) \wedge (y \in a^*) \wedge (z \in b^*) \wedge \varphi_{\text{Add}}(x, y, z)), \\ \psi_4 &:= \exists \mathbf{u}, x, y, z: (\varphi_w(\mathbf{u}) \wedge (\mathbf{u} \doteq xyz) \wedge (x \in b^*) \wedge (y \in a^*) \wedge (z \in b^*) \wedge \varphi_{\text{Mult}}(x, y, z)), \\ \psi_5 &:= \exists \mathbf{u}, x, y: (\varphi_w(\mathbf{u}) \wedge (\mathbf{u} \doteq xy) \wedge (x \in (abaabb)^*) \wedge (y \in (bbaaba)^*) \wedge \varphi_{\text{Perm}}(x, y)), \\ \psi'_5 &:= \exists \mathbf{u}, x, y: (\varphi_w(\mathbf{u}) \wedge (\mathbf{u} \doteq xy) \wedge (x \in (abaabb)^*) \wedge (y \in (bbaaba)^*) \wedge \varphi_{\text{Rev}}(x, y)), \\ \psi_6 &:= \exists \mathbf{u}, x, y, z: (\varphi_w(\mathbf{u}) \wedge (\mathbf{u} \doteq xyz) \wedge (x \in a^+) \wedge (y \in b^+) \wedge \varphi_{\text{Shuff}}(x, y, z)). \end{aligned}$$

It follows that $\mathcal{L}(\psi_i) = L_i$ for $i \in [6]$, where L_i is defined in Lemma 4.16, and $\mathcal{L}(\psi'_5) = L_5$. Thus the relations $\text{Num}_a, \text{Add}, \text{Mult}, \text{Scatt}, \text{Perm}, \text{Rev}$, and Shuff are not definable in $\text{FC}[\text{REG}]$.

Now, let $\Sigma := \{a, b\}$. To conclude the proof, assume that for the morphism $h: \Sigma^* \rightarrow \Sigma^*$, where $h(a) = b$ and $h(b) = b$, the relation Morph_h is definable in $\text{FC}[\text{REG}]$. Then, let

$$\psi := \exists \mathbf{u}, x, y: (\varphi_w(\mathbf{u}) \wedge (\mathbf{u} \doteq x \cdot y) \wedge (x \in a^*) \wedge \text{Morph}_h(x, y)).$$

Thus, $\mathcal{L}(\psi) = \{a^n b^n \mid n \in \mathbb{N}\}$, and therefore $\mathcal{L}(\psi) \notin \mathcal{L}(\text{FC}[\text{REG}])$ (see [12] or Example 4.5 along with Lemma 5.3). Thus, Morph_h is not definable in $\text{FC}[\text{REG}]$. $\square$

Immediately from [12], all the relations given in Theorem 5.6 are not selectable by generalized core spanners. Consequently, if one needed to express a query from Theorem 5.6, they would have to add a specialized operator to obtain this functionality.

PROPOSITION 5.7. *Any relation $R \subseteq (\Sigma^*)^k$, for some $k \in \mathbb{N}$, is definable in $\text{FC}[\text{REG}]$ if and only if $(\Sigma^*)^k \setminus R$ is definable in $\text{FC}[\text{REG}]$.*

This is simply due to the fact that FC is closed under complementation. Thus, the complements of all relations given in Theorem 5.6 are also not selectable by generalized core spanners.

Core Spanners. Core spanners are a subclass of generalized core spanners (see [6]). While there are known inexpressibility results for core spanners (for example, Num_a , Scatt , Perm , Rev , and $R_{<} := \{(u, v) \mid |u| < |v|\}$ were shown to be not selectable by core spanners in [9], and [6] proved that length equality is not selectable by core spanners), our results extend what is known about the expressive power of core spanners. For example, since the complement of the relations given in Theorem 5.6 are not selectable by generalized core spanners, they are also not selectable by core spanners. Furthermore, we also show that Add , Mult , Shuff and Morph_h are not selectable by generalized core spanners, and thus are not selectable by core spanners.

From [12], the existential-positive fragment of $\text{FC}[\text{REG}]$ corresponds to the core spanners (in the same way that $\text{FC}[\text{REG}]$ corresponds to the generalized core spanners). Since existential Ehrenfeucht-Fraïssé games can give inexpressibility results for existential-positive first-order logic (see Theorem 2.2 of [1]), this seems like a promising direction for core spanner inexpressibility.

6 CONCLUSIONS

In Section 3, we established Ehrenfeucht-Fraïssé games for FC. Then, in Section 4, we gave two technical lemmas – the Pseudo-Congruence Lemma, and the Primitive Power Lemma – which serve the basis of the Fooling Lemma. These lemmas are the main contributions of this work; however, in Section 5, we lifted our inexpressibility results to $\text{FC}[\text{REG}]$ which immediately from [12] yields inexpressibility results for generalized core spanners.

The main limitation of our results is that our generalized core spanner inexpressibility results rely on Boolean combinations of bounded languages. It is possible to use closure properties in order to partially overcome this issue. For example, consider the language $L := \{w \in \Sigma^* \mid |w|_a = |w|_b\}$. Since $\text{FC}[\text{REG}]$ is closed under intersection with regular languages, $L \in \mathcal{L}(\text{FC}[\text{REG}])$ if and only if $L \cap a^*b^* \in \mathcal{L}(\text{FC}[\text{REG}])$. However, $L \cap a^*b^*$ is the language $a^n b^n$ and thus $L \notin \mathcal{L}(\text{FC}[\text{REG}])$.

Such an approach can only get us so far, and therefore going beyond Boolean combinations of bounded languages is an important area for future research. We note that it is still open whether FC and $\text{FC}[\text{REG}]$ have the same expressive power. The authors believe that this is unlikely, although proving that FC and $\text{FC}[\text{REG}]$ do indeed have equivalent expressive power would remove the restriction of Boolean combinations of bounded languages.

As an intermediate step, one could look at more general regular languages that can be expressed by FC. This would close the gap between what we know can be expressed by FC and what can be expressed by $\text{FC}[\text{REG}]$. Then, a similar result to Lemma 5.3 could gain further inexpressibility results for generalized core spanners. A starting point for this direction would be to see whether regular constraints beyond the Boolean closure of simple regular expressions are achievable in FC.

Another promising area of future work, is looking at related two player games. For example, pebble games could be used to derive insights on FC-formulas with a finite number of variables (for example, see Chapter 11 of [21]). Alternatively, the restriction to existential Ehrenfeucht-Fraïssé games could yield further results regarding the inexpressibility of core spanners.

We note that there are many further open problems regarding FC. We refer to [12] for a more comprehensive list of future research directions.

DATA AVAILABILITY

No data was analysed, captured or generated for this work.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable feedback. This work was funded by EPSRC grant EP/T033762/1.

REFERENCES

- [1] Samson Abramsky and Nihil Shah. 2021. Relating structure and power: Comonadic semantics for computational resources. *Journal of Logic and Computation* 31, 6 (2021), 1390–1428.
- [2] Antoine Amarilli, Pierre Bourhis, Stefan Mengel, and Matthias Niewerth. 2020. Constant-delay enumeration for nondeterministic document spanners. *ACM SIGMOD Record* 49, 1 (2020), 25–32.
- [3] Laura Chiticariu, Rajasekar Krishnamurthy, Yunyao Li, Sriram Raghavan, Frederick Reiss, and Shivakumar Vaithyanathan. 2010. SystemT: An algebraic approach to declarative information extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. 128–137.
- [4] Heinz-Dieter Ebbinghaus and Jörg Flum. 1999. *Finite model theory*. Springer Science & Business Media.
- [5] Ronald Fagin. 1996. Easier ways to win logical games. *Descriptive complexity and finite models* 31 (1996), 1–32.
- [6] Ronald Fagin, Benny Kimelfeld, Frederick Reiss, and Stijn Vansummeren. 2015. Document Spanners: A Formal Approach to Information Extraction. *Journal of the ACM* 62, 2 (2015), 12.
- [7] Nathan J Fine and Herbert S Wilf. 1965. Uniqueness theorems for periodic functions. *Proc. Amer. Math. Soc.* 16, 1 (1965), 109–114.
- [8] Fernando Florenzano, Cristian Riveros, Martín Ugarte, Stijn Vansummeren, and Domagoj Vrgoč. 2020. Efficient enumeration algorithms for regular document spanners. *ACM Transactions on Database Systems (TODS)* 45, 1 (2020), 1–42.
- [9] Dominik D. Freydenberger. 2019. A Logic for Document Spanners. *Theory of Computing Systems* 63, 7 (2019), 1679–1754.
- [10] Dominik D. Freydenberger and Mario Holldack. 2018. Document Spanners: From Expressive Power to Decision Problems. *Theory of Computing Systems* 62, 4 (2018), 854–898.
- [11] Dominik D Freydenberger and Liat Peterfreund. 2019. The theory of concatenation over finite models. *arXiv preprint arXiv:1912.06110* (2019).
- [12] Dominik D. Freydenberger and Liat Peterfreund. 2021. The theory of concatenation over finite models. In *Proceedings of IICALP 2021*. 130:1–130:17.
- [13] Dominik D. Freydenberger and Sam M. Thompson. 2022. Splitting Spanner Atoms: A Tool for Acyclic Core Spanners. In *Proceedings of ICDT 2022*. 6:1–6:18.
- [14] Seymour Ginsburg and Edwin Spanier. 1966. Semigroups, Presburger formulas, and languages. *Pacific J. Math.* 16, 2 (1966), 285–296.
- [15] Jana Hadravová and Štěpán Holub. 2015. Equation $x^i y^j x^k = u^i v^j u^k$ in Words. In *Language and Automata Theory and Applications: 9th International Conference, LATA 2015, Nice, France, March 2-6, 2015, Proceedings*. Springer, 414–423.
- [16] Neil Immerman. 1998. *Descriptive complexity*. Springer Science & Business Media.
- [17] Juhani Karhumäki. 1983. On cube-free ω -words generated by binary morphisms. *Discrete Applied Mathematics* 5, 3 (1983), 279–297.
- [18] Juhani Karhumäki, Filippo Mignosi, and Wojciech Plandowski. 2000. The expressibility of languages and relations by word equations. *Journal of the ACM* 47, 3 (2000), 483–505.
- [19] Rajasekar Krishnamurthy, Yunyao Li, Sriram Raghavan, Frederick Reiss, Shivakumar Vaithyanathan, and Huaiyu Zhu. 2009. SystemT: a system for declarative information extraction. *ACM SIGMOD Record* 37, 4 (2009), 7–13.
- [20] Yunyao Li, Frederick Reiss, and Laura Chiticariu. 2011. SystemT: A declarative information extraction system. In *Proceedings of the ACL-HLT 2011 System Demonstrations*. 109–114.
- [21] Leonid Libkin. 2004. *Elements of finite model theory*. Springer.
- [22] M. Lothaire. 1997. *Combinatorics on words*. Vol. 17. Cambridge university press.
- [23] M. Lothaire. 2002. *Algebraic combinatorics on words*. Vol. 90. Cambridge university press.
- [24] Johann A Makowsky. 2004. Algorithmic uses of the Feferman–Vaught theorem. *Annals of Pure and Applied Logic* 126, 1–3 (2004), 159–213.
- [25] Liat Peterfreund. 2021. Grammars for Document Spanners. In *Proceedings of ICDT 2021*. 7:1–7:18.
- [26] Liat Peterfreund, Balder ten Cate, Ronald Fagin, and Benny Kimelfeld. 2019. Recursive Programs for Document Spanners. In *Proceedings of ICDT 2019*. 13:1–13:18.
- [27] Grzegorz Rozenberg and Arto Salomaa. 1997. *Handbook of Formal Languages, Volume 1: Word, Language, Grammar*. Springer.
- [28] Markus L. Schmid and Nicole Schweikardt. 2021. A Purely Regular Approach to Non-Regular Core Spanners. In *Proceedings of ICDT 2021*. 4:1–4:19.

- [29] Markus L. Schmid and Nicole Schweikardt. 2021. Spanner evaluation over SLP-compressed documents. In *Proceedings of PODS 2021*. 153–165.
- [30] Wolfgang Thomas. 1993. On the Ehrenfeucht-Fraïssé game in theoretical computer science. In *Colloquium on Trees in Algebra and Programming*. Springer, 559–568.
- [31] Sam M Thompson and Dominik D Freydenberger. 2023. Generalized Core Spanner Inexpressibility via Ehrenfeucht-Fraïssé Games for FC. *arXiv preprint arXiv:2306.16364* (2023).
- [32] Sam M. Thompson and Dominik D. Freydenberger. 2023. Languages Generated by Conjunctive Query Fragments of FC[REG]. In *Proceedings of DLT 2023*. 233–245.

Received June 2023; revised August 2023; accepted September 2023