Games for Succinctness of Regular Expressions

Miikka Vilander
Computing Sciences
Tampere University
Tampere, Finland
miikka.vilander@tuni.fi

We present a version of so called formula size games for regular expressions. These games characterize the equivalence of languages up to expressions of a given size. We use the regular expression size game to give a simple proof of a known non-elementary succinctness gap between first-order logic and regular expressions. We also use the game to only count the number of stars in an expression instead of the overall size. For regular expressions this measure trivially gives a hierarchy in terms of expressive power. We obtain such a hierarchy also for what we call RE over star-free expressions, where star-free expressions, that is ones with complement but no stars, are combined using the operations of regular expressions.

1 Introduction

Even though regular expressions, abbreviated RE, are a very thoroughly studied topic in computer science, little work has been done on their succinctness, or size, until recently. The pioneering paper on the size of RE seems to be in 1974 by Ehrenfeucht and Zeiger [4]. They define the size of an RE as the number of occurrences of alphabet symbols in it and show that there is a deterministic finite automata with n states such that the smallest RE defining the same language has size 2^{n-1}. In 2005, Ellul et al. [5] noted the lack of work on succinctness and presented several open problems as well as some results of their own. Some of these open problems were related to the succinctness of RE expanded with operations such as intersection. These and other similar problems were independently solved by Gelade and Neven [6, 7] on the one hand and Gruber and Holzer [8, 9] on the other.

Gelade and Neven use a generalization of the result of Ehrenfeucht and Zeiger [4] to obtain double exponential lower bounds for the size of an RE defining the complement of a single RE or the intersection of a finite number of RE in a fixed size alphabet [7]. Gelade uses the same technique to also obtain double exponential lower bounds for the added operations of interleaving and counting [6]. Gruber and Holzer go even further, obtaining tighter bounds for all of the above in a two-letter alphabet [8, 9]. They link the size of RE to their star height via a measure on the connectivity of the underlying DFA. The measure is called cycle rank and was first introduced by Eggan and Büchi [3]. These two groups worked independently although they were clearly aware of the other group's work.

Many problems in finite model theory have been solved via the use of games such as the famous Ehrenfeucht-Fraïssé game that characterizes quantifier rank or depth in first-order logic. A similar game for RE was presented by Yan [15]. This so called split game characterizes the depth of both catenation and stars for generalized regular expressions, or GRE, where complement is added as an operation. Catenation depth is sometimes referred to as dot-depth and star depth is more commonly known as star height. For RE, Hashiguchi famously proved that star height gives a full hierarchy in terms of expressive power [10]. For GRE, it is notoriously not even known if a language that requires an expression of star height two exists. Yan offers his game as a possible way to attack the generalized star height problem but is only able to complete results on infinite ω-words.

D. Bresolin and P. Ganty (Eds.): 12th International Symposium
on Games, Automata, Logics, and Formal Verification (GandALF 2021)
EPTCS 346, 2021, pp. 258–272, doi:10.4204/EPTCS.346.17

In the vein of EF-games, there are also games for succinctness. These are often called formula size games. They are games of definability just as the EF-game, but instead of quantifier rank they measure the size of the defining formula. To our knowledge, the earliest example of such a game is for propositional logic by Razborov [13]. Perhaps more well known is the later game by Adler and Immerman [1] for a modal logic called CTL. To our knowledge, ours are the first formula size games presented for regular expressions.

While EF-games are played on two structures, formula size games are instead played on two sets of structures, A and B. In the context of regular expressions, these sets are languages. Our version of the games also has a resource parameter k. The first player S is trying to show that there is an expression R with $A \subseteq L(R)$, $B \subseteq \Sigma^* \setminus L(R)$ and size at most k. S essentially sketches the syntax tree of such a separating expression as the game goes on, but in a single game only one branch of the tree is visited. It is the role of the second player D to choose which branch this is, and try to find the error in the strategy of S. A separating expression of appropriate size exists if and only if S has a winning strategy. In addition to the size, in this paper we are also interested in the number of stars in an expression. Thus we add a separate parameter s to the game to track this. The game is very easy to modify in this way to track the number or depth of whatever operators one is interested in.

We use the RE-version of the game to give a simpler proof for a known non-elementary succinctness gap between FO and RE. Stockmeyer [14] showed that star-free expressions are non-elementarily more succinct than RE and together with an elementary translation from FO to star-free by McNaughton and Papert [12], the result follows. In addition, we consider the number of stars in an expression as a measure of complexity. For RE a hierarchy in terms of expressive power can be trivially obtained in star height one. For GRE this presents a difficult problem as the full use of complement ramps up the complexity of the game significantly. We present RE over star-free expressions as a natural middle ground between RE and GRE. These include all star-free expressions with complement and their combinations using the operations of RE. For RE over star-free expressions we use a corresponding version of the game to show that the number of stars also gives a full hierarchy in terms of expressive power already in star height one.

The outline of the paper is as follows. In Section 2 we introduce RE, GRE and RE over star-free expressions. We also discuss our definition of size for these expressions and define some notation for the rest of the paper. In Section 3 we present the GRE size game and prove that it works as intended. We also present variations of the game for RE and RE over star-free, and prove some useful lemmas for later. In Section 4 we use the game for RE to show that defining a large finite language requires a large RE. We then define a finite language of non-elementary size via a FO-formula of exponential size, thus reproving the succinctness gap between FO and RE. In Section 5 we show that the number of stars in an expression gives a hierarchy in terms of expressive power for RE over star-free expressions. We conclude in Section 6.

2 Preliminaries

We begin by defining some basic notions such as regular expressions and our concept of the size of a regular expression. For more on regular expressions we refer the reader to [11]. We omit the syntax and semantics of first-order logic and direct the reader to [2] for a textbook with a finite model theory approach.

Let Σ be an alphabet. Strings of symbols from the alphabet are called *words* and sets of words are called *languages*. We denote the length of a word w with $|w|$.

The *regular expressions*, or RE, of Σ are defined recursively as follows: $\emptyset$, ε and every $a \in \Sigma$ are regular expressions. If R_1 and R_2 are regular expressions, then also $R_1 \cup R_2$, R_1R_2 and R_1^* are regular expressions. The *generalized regular expressions*, or GRE, of Σ are defined in the same way with the following addition: if R is a GRE, then $\neg R$ is also a GRE. Sometimes GRE are also defined to include a separate intersection operation. As the effect on succinctness is negligible, we define intersection as the shorthand $R_1 \cap R_2 := \neg(\neg R_1 \cup \neg R_2)$ to keep the number of moves in our game smaller.

The *language of a regular expression* R, denoted by $L(R)$ is defined as follows:

- $L(\emptyset) = \emptyset$,
- $L(\varepsilon) = \{\varepsilon\}$ (the empty word),
- $L(a) = \{a\}$ for $a \in \Sigma$,
- $L(R_1 \cup R_2) = L(R_1) \cup L(R_2)$,
- $L(R_1R_2) = L(R_1)L(R_2) = \{uv \mid u \in L(R_1), v \in L(R_2)\}$ and
- $L(R_1^*) = L(R_1)^* = \{w_1 \cdots w_n \mid n \in \mathbb{N}, w_i \in L(R_1) \text{ for each } i \in \mathbb{N}\}$.

For generalized regular expressions, additionally $L(\neg R_1) = \Sigma^* \setminus L(R_1)$.

We will also refer to *star-free expressions*. These are generalized regular expressions with the $*$-rule removed. A classical result by McNaughton and Papert [12] states that star-free expressions have the same expressive power over words as first-order logic. Note that this means many languages naturally expressed by a RE with stars are also expressible by star-free expressions. For example, if $\Sigma = \{a, b\}$, then $L((ab)^*) = L(\varepsilon \cup (a\neg\emptyset \cap \neg\emptyset b \cap \neg(\neg\emptyset aa\neg\emptyset) \cap \neg(\neg\emptyset bb\neg\emptyset)))$.

Finally we present a middle ground between RE and GRE we call *RE over star-free expressions*. These expressions are defined by R in the following grammar (we omit parentheses for simplicity):

$$R ::= R \cup R \mid RR \mid R^* \mid S$$
$$S ::= S \cup S \mid SS \mid \neg S \mid \emptyset \mid \varepsilon \mid a \text{ for every } a \in \Sigma$$

As the name suggests, RE over star-free expressions include all star-free expressions in the sense of GRE and can combine them using only the operations of RE. Essentially this means that stars cannot occur inside a complement. Since star-free expressions correspond to FO-definable properties of words, we feel this is a natural variation of RE to consider in terms of succinctness. It is quite possible someone else has already presented it but we could not find it in the literature.

There are several ways one could define the size of a regular expression. Gruber and Holzer [8] use alphabetic width defined as the number of occurrences of symbols from Σ in the expression. Gelade and Neven [7] on the other hand note that this is not sufficient for GRE since one can construct non-trivial expressions with no symbols from Σ. Thus they count also operations, ending up with the size of the syntax tree of the expression. This is also sometimes called *reverse polish length* [5]. We use the latter concept here but the game can easily be adapted to alphabetic width or actual string length with parentheses if desired.

Definition 2.1. The *size* of a GRE is defined recursively as follows:

- $\mathrm{sz}(\emptyset) = \mathrm{sz}(\varepsilon) = \mathrm{sz}(a) = 1$ for every $a \in \Sigma$,
- $\mathrm{sz}(R^*) = \mathrm{sz}(\neg R) = \mathrm{sz}(R) + 1$ and
- $\mathrm{sz}(R_1 \cup R_2) = \mathrm{sz}(R_1R_2) = \mathrm{sz}(R_1) + \mathrm{sz}(R_2) + 1$.

In the sequel we will deal with some rather large expression sizes. In particular, we will show a *non-elementary* succinctness gap between FO and RE. This means that the difference in required size is not expressible by an elementary function. In practice, it suffices to show that the size of the RE is above an exponential tower. For this, we define the function twr as follows:

- $\mathrm{twr}(0) = 1$,
- $\mathrm{twr}(n+1) = 2^{\mathrm{twr}(n)}$.

We also use the shorthand

$$[n] := \{1, \ldots, n\}.$$

Finally we define some concepts and notations for the RE size game. First is the concept of regular expressions separating languages.

Definition 2.2. Let $A, B \subseteq \Sigma^*$. A GRE R *separates* A *from* B if $A \subseteq L(R)$ and $B \subseteq \Sigma^* \setminus L(R)$.

Note that if $A = L(R)$ and $B = \Sigma^* \setminus L(R)$, then R defines the language A, so separation is a sort of partial version of defining languages with expressions.

To consider catenation and star in the game, we will need notation for the different ways one can split a word into two or more shorter words.

Let $w \in \Sigma^*$ and $n \in \mathbb{N}$. The set of *n-splits of* w is the set

$$\mathrm{Sp}^n(w) = \{(w_1, \ldots, w_n) \mid w_1 \ldots w_n = w\}.$$

We also use the notation

$$\mathrm{Sp}(w) := \bigcup_{n \in \mathbb{N}} \mathrm{Sp}^n(w)$$

for the set of all splits of w.

3 Generalized regular expression size game

In this section we define a game for generalized regular expressions that is the equivalent of so called formula size games previously developed for different logics. Since we consider both overall size and number of stars in this paper, we present a game with a separate parameter for stars.

The GRE size game has two players, Samson (S) and Delilah (D). The game has four parameters: two sets of Σ-words, A_0 and B_0, and two natural numbers k_0 and s_0 with $k_0 \geq s_0$. Samson wants to show that A_0 can be separated from B_0 using a GRE with size at most k_0 and at most s_0 stars. Delilah wants to refute this. The GRE size game with the above parameters is denoted by $\mathrm{GRES}(k_0, s_0, A_0, B_0)$.

Positions of the game are of the form (k, s, A, B) where A and B are sets of words, $k, s \in \mathbb{N}$ and $k \geq s$. The starting position is (k_0, s_0, A_0, B_0). In a position $P = (k, s, A, B)$, if $k = 0$, then the game ends and D wins. Otherwise S has a choice of six moves (note that the empty word ε is covered in the a-move):

- a-move: S chooses $a \in \Sigma \cup \{\varepsilon\}$. If $A \subseteq \{a\}$ and $a \notin B$, the game ends and S wins. Otherwise D wins.
- $\emptyset$-move: If $A = \emptyset$, S wins. Otherwise D wins.
- $\cup$-move: S chooses subsets $A_1, A_2 \subseteq A$ such that $A_1 \cup A_2 = A$ and natural numbers k_1, k_2, s_1, s_2 such that $k_i \geq s_i$, $k_1 + k_2 + 1 = k$ and $s_1 + s_2 = s$. Then D chooses a number $i \in \{1, 2\}$. The game continues from the position (k_i, s_i, A_i, B).

- cat-move: For every $w \in A$, S chooses a 2-split (w_1, w_2). Let $A_i = \{w_i \mid w \in A\}$. Then for every $v \in B$, S chooses a function $f_v : \mathrm{Sp}^2(v) \to \{1,2\}$. Let $B_i = \{v_i \mid f_v(v_1, v_2) = i, (v_1, v_2) \in \mathrm{Sp}^2(v)\}$. S chooses numbers k_1, k_2, s_1, s_2 such that $k_i \geq s_i$, $k_1 + k_2 + 1 = k$ and $s_1 + s_2 = s$. Finally D chooses a number $i \in \{1,2\}$. The game continues from the position (k_i, s_i, A_i, B_i).
- $*$-move: If $\varepsilon \in B$, D wins. Otherwise, for every $w \in A \setminus \{\varepsilon\}$, S chooses a natural number $n(w) > 0$ and an $n(w)$-split $(w_1, \ldots, w_{n(w)})$ with $w_i \neq \varepsilon$ for every $i \in [n(w)]$. Let $A' = \{w_i \mid i \in [n(w)], w \in A\}$. Then for every $v \in B$, S chooses a function $f_v : \mathrm{Sp}(v) \to \mathbb{N}$ such that $f_v(v_1, \ldots, v_n) \in [n]$. Let $B' = \{v_i \mid f_v(v_1, \ldots, v_n) = i, (v_1, \ldots, v_n) \in \mathrm{Sp}(v)\}$. The game continues from the position $(k-1, s-1, A', B')$.
- $\neg$-move: The game continues from the position $(k-1, s, B, A)$.

Note that since every move either ends the game or decreases the resource k, the game always ends in a finite number of moves and one of the players wins.

We now prove the crucial theorem that states the connection of the game to the succinctness of generalized regular expressions.

Theorem 3.1. *Let $A, B \subseteq \Sigma^*$ and $k, s \in \mathbb{N}$ with $k \geq s$. The following are equivalent:*

1. *S has a winning strategy in the game* $\mathrm{GRES}(k, s, A, B)$.
2. *There is a generalized regular expression that separates A from B with size at most k and at most s stars.*

Proof. In the following we will always have $i \in \{1,2\}$ without explicit statement. We show the equivalence of 1 and 2 for all A and B by induction on the number k. The case $k = 0$ is clear.

$1 \Rightarrow 2$: Let δ be a winning strategy for S in the game $\mathrm{GRES}(k, A, B)$. Since δ is a winning strategy, we have $k > 0$. The proof is divided into cases according to the first move of δ:

- a-move: If the first move is an a-move, because δ is a winning strategy, we have $A \subseteq \{a\} = L(a)$ and $a \notin B$ so $B \subseteq \Sigma^* \setminus L(a)$. Thus the regular expression a separates A from B.
- $\emptyset$-move: Now $A = \emptyset$ so $\emptyset$ separates A from B.
- $\cup$-move: S chooses $A_1, A_2 \subseteq A$ and k_1, k_2, s_1, s_2 according to δ. Since δ is a winning strategy, S has winning strategies from both of the possible following positions (k_i, s_i, A_i, B). Thus by induction hypothesis there are GREs R_1 and R_2 such that R_i separates A_i from B, $\mathrm{sz}(R_i) \leq k_i$ and R_i has at most s_i stars. Now $A_i \subseteq R_i$ and $B \subseteq \Sigma^* \setminus L(R_i)$. Therefore

$$A_0 = A_1 \cup A_2 \subseteq L(R_1) \cup L(R_2) = L(R_1 \cup R_2).$$

  and $B \subseteq (\Sigma^* \setminus L(R_1)) \cap (\Sigma^* \setminus L(R_2)) = \Sigma^* \setminus L(R_1 \cup R_2)$ so $R_1 \cup R_2$ separates A from B. In addition, $\mathrm{sz}(R_1 \cup R_2) = \mathrm{sz}(R_1) + \mathrm{sz}(R_2) + 1 \leq k_1 + k_2 + 1 = k$ and $R_1 \cup R_2$ has at most $s_1 + s_2 = s$ stars.
- cat-move: S makes his choices according to δ. Now S has a winning strategy for both positions (k_i, s_i, A_i, B_i) so by induction hypothesis there are GREs R_1 and R_2 such that R_i separates A_i from B_i, $\mathrm{sz}(R_i) \leq k_i$ and R_i has at most s_i stars. Now $A_i \subseteq L(R_i)$. For every $w \in A$ there are $w_1 \in A_1$ and $w_2 \in A_2$ such that $w_1 w_2 = w$ so $A \subseteq L(R_1)L(R_2) = L(R_1 R_2)$. On the other side $B_i \subseteq \Sigma^* \setminus L(R_i)$. For every $v \in B$ and every $(v_1, v_2) \in \mathrm{Sp}^2(v)$, either $v_1 \in B_1$ or $v_2 \in B_2$. Thus $v \notin L(R_1)L(R_2) = L(R_1 R_2)$ so $B \subseteq \Sigma^* \setminus L(R_1 R_2)$. The GRE $R_1 R_2$ thus separates A from B. The size and number of stars are handled as in the previous case.

- $*$-move: S makes his choices according to δ. S has a winning strategy for the following position $(k-1, s-1, A', B')$ so by induction hypothesis there is a GRE R such that R separates A' from B', $\mathrm{sz}(R) \leq k-1$ and R has at most $s-1$ stars. We have $A' \subseteq L(R)$. For every $w \in A$ there is $n(w) \in \mathbb{N}$ and an $n(w)$-split $(w_1, \ldots, w_{n(w)})$ such that $w_j \in A'$ for $j \in [n(w)]$. Thus $A \subseteq L(R)^* = L(R^*)$. On the other side, $B' \subseteq \Sigma^* \setminus L(R)$. For every $v \in B$ and every $(v_1, \ldots, v_n) \in \mathrm{Sp}(v)$, there is $j \in [n]$ such that $v_j \in B'$. Thus $v \notin L(R)^* = L(R^*)$ so $B \subseteq \Sigma \setminus L(R^*)$. The GRE R^* thus separates A from B. In addition, $\mathrm{sz}(R^*) = \mathrm{sz}(R) + 1 \leq k$ and R^* has at most $s - 1 + 1 = s$ stars.
- $\neg$-move: S has a winning strategy from the following position $(k-1, s, B, A)$ so there is a GRE R that separates B from A with $\mathrm{sz}(R) \leq k-1$ and at most s stars. Now the GRE $\neg R$ separates A from B. In addition, $\mathrm{sz}(\neg R) = \mathrm{sz}(R) + 1 \leq k$ and $\neg R$ has at most s stars.

$2 \Rightarrow 1$: Let R be a GRE that separates A and B with size at most k and at most s stars. The proof is divided into cases according to the outermost operator in R:

- $R = a \in \Sigma \cup \{\varepsilon\}$: Since R separates A from B, we have $A \subseteq \{a\}$ and $B \subseteq \Sigma^* \setminus \{a\}$ so $a \notin B$. Thus S wins by making an a-move.
- $R = \emptyset$: Now $A = \emptyset$ so S wins by making a $\emptyset$-move.
- $R = R_1 \cup R_2$: Since R separates A from B, we have $A \subseteq L(R) = L(R_1) \cup L(R_2)$. Let $A_i = A \cap L(R_i)$, let $k_1 = \mathrm{sz}(R_1)$ and let $k_2 = k - k_1 - 1$. Similarly let s_1 be the number of stars in R_1 and let $s_2 = s - s_1$. Now $A_1 \cup A_2 = A$, $k_i > s_i$, $k_1 + k_2 + 1 = k$ and $s_1 + s_2 = s$ so these are valid choices for a $\cup$-move. After the $\cup$-move, $A_i \subseteq L(R_i)$ and $B \subseteq \Sigma^* \setminus L(R) = (\Sigma^* \setminus L(R_1)) \cap (\Sigma^* \setminus L(R_2))$ so $B \subseteq \Sigma^* \setminus L(R_i)$. Now R_i separates A_i from B. In addition, $\mathrm{sz}(R_1) = k_1$, $\mathrm{sz}(R_2) = \mathrm{sz}(R) - \mathrm{sz}(R_1) - 1 \leq k - k_1 - 1 = k_2$. Similarly R_1 has s_1 stars and R_2 has at most $s - s_1 = s_2$ stars. By induction hypothesis, S has a winning strategy for the game $\mathrm{GRES}(k_i, s_i, A_i, B)$. Together with the first move, this is a winning strategy for the game $\mathrm{GRES}(k, s, A, B)$.
- $R = R_1 R_2$: Since R separates A from B, we have $A \subseteq L(R) = L(R_1)L(R_2)$. Thus for every $w \in A_0$ there is $(w_1, w_2) \in \mathrm{Sp}^2(w)$ such that $w_1 \in L(R_1)$ and $w_2 \in L(R_2)$. S makes a cat-move and chooses such a split for each $w \in A$. On the other side we have $B \subseteq \Sigma^* \setminus L(R) = \Sigma^* \setminus L(R_1)L(R_2)$. Thus for every $v \in B$ and every $(v_1, v_2) \in \mathrm{Sp}^2(v)$, we have $v_1 \notin L(R_1)$ or $v_2 \notin L(R_2)$. For the function $f_v : \mathrm{Sp}(v) \to \mathbb{N}$, S chooses $i = f_v(v_1, v_2)$ so that $v_i \notin L(R_i)$. S chooses k_i and s_i as in the previous case. Finally we have $A_i \subseteq L(R_i)$ and $B_i \subseteq \Sigma^* \setminus L(R_i)$ so R_i separates A_i from B_i. The resources k and s are handled like in the previous case. By induction hypothesis, S has a winning strategy from the position (k_i, s_i, A_i, B_i).
- $R = R_1^*$: Since R separates A from B, we have $A \subseteq L(R) = L(R_1)^*$. Thus for every $w \in A$ there is $(w_1, \ldots, w_n) \in \mathrm{Sp}(w)$ such that $w_j \in L(R_1)$ for all $j \in [n]$. S makes a $*$-move and chooses such a split for each $w \in A$. On the other side we have $B \subseteq \Sigma^* \setminus L(R) = \Sigma^* \setminus L(R_1)^*$. Note that $\varepsilon \notin B$ so D does not win outright. Now for every $v \in B$ and every $(v_1, \ldots, v_n) \in \mathrm{Sp}(v)$ we have $v_j \notin L(R_1)$ for some $j \in [n]$. For the function $f_v : \mathrm{Sp}(v) \to \mathbb{N}$, S chooses $j = f_v(v_1, \ldots, v_n)$ so that $v_j \notin L(R_1)$. Finally we have $A' \subseteq L(R_1)$ and $B' \subseteq \Sigma^* \setminus L(R_1)$ so R_1 separates A' from B'. In addition, $\mathrm{sz}(R_1) = \mathrm{sz}(R) - 1 \leq k - 1$ and R_1 has at most $s - 1$ stars. By induction hypothesis, S has a winning strategy from the position $(k-1, s-1, A', B')$.
- $R = \neg R_1$: S makes a $\neg$-move. Since R separates A from B, it follows that R_1 separates B from A. In addition, $\mathrm{sz}(R_1) = \mathrm{sz}(R) - 1 \leq k - 1$ and R_1 has at most s stars. By induction hypothesis, S has a winning strategy from the position $(k-1, s, B, A)$.

□

We have defined the game for generalized regular expressions but this full game turns out to be very complex in a combinatorial sense. For the results in this paper we will use simpler games for RE and RE over star-free.

The RE size game $\mathrm{RES}(k,A,B)$ is the game $\mathrm{GRES}(k,s,A,B)$ with the $\neg$-move and the star parameter s removed. The proof of Theorem 3.1 with the $\neg$-move cases and s removed proves the following analogue for this game:

Theorem 3.2. *Let $A,B \subseteq \Sigma^*$, $k \in \mathbb{N}$. The following are equivalent:*

1. *S has a winning strategy in the game* $\mathrm{RES}(k,A,B)$.
2. *There is a regular expression that separates A from B with size at most k.*

The RE over star-free size game $\mathrm{RESFS}(k,s,A,B)$ is the game $\mathrm{GRES}(k,s,A,B)$ with the following modification: after a $\neg$-move, the following position is $(k,0,B,A)$ instead of the normal (k,s,B,A). This corresponds with the syntax of RE over star-free, where stars cannot occur under complement. We omit the proof of the analogous theorem for this game:

Theorem 3.3. *Let $A,B \subseteq \Sigma^*$ and $k,s \in \mathbb{N}$ with $k \geq s$. The following are equivalent:*

1. *S has a winning strategy in the game* $\mathrm{RESFS}(k,s,A,B)$.
2. *There is a RE over star-free expression that separates A from B with size at most k and at most s stars.*

As is usual with these sorts of games, we will need a simple lemma stating that if the same word is present on both sides of the game, D has a winning strategy. We prove the lemma for the GRE game and note that it can just as easily be proven for the other variations.

Lemma 3.4. *In a position $P = (k,s,A,B)$ of a game* $\mathrm{GRES}(k_0,s_0,A_0,B_0)$*, if there is $w \in A \cap B$, then D has a winning strategy from position P.*

Proof. Under the assumptions, we describe a strategy for D. For any move of S, this strategy either wins or maintains the condition of having $w \in A \cap B$. It is thus a winning strategy. We consider the cases for each possible move of S.

- a-move: Assume S chooses $a \in \Sigma \cup \{\varepsilon\}$. If $A \subseteq \{a\}$, then $a = w \in B$, so D wins.
- $\emptyset$-move: Since $w \in A$, $A \neq \emptyset$ and D wins.
- $\cup$-move: Assume S chooses subsets $A_1, A_2 \subseteq A$. Since $A_1 \cup A_2 = A$, there is $i \in \{1,2\}$ such that $w \in A_i$. D chooses this i and in the following position (k_i,s_i,A_1,B), we have $w \in A_i \cap B$.
- cat-move: Let (w_1,w_2) be the split S chooses for w on the A-side and let $f_w : \mathrm{Sp}^2(w) \to \{1,2\}$ be the function S chooses for w on the B-side. D chooses the number $i := f_w(w_1,w_2)$. In the following position (k_i,s_i,A_i,B_i), we have $w_i \in A_i \cap B_i$.
- $*$-move: If $w = \varepsilon$, D wins. Otherwise, let $(w_1,\ldots,w_n)$ be the split S chooses for w on the A-side and let $f_w : \mathrm{Sp}(w) \to \mathbb{N}$ be the function S chooses for w on the B-side. Let $i := f_w(w_1,\ldots,w_n)$. In the following position $(k-1,s-1,A',B')$ we have $w_i \in A' \cap B'$.
- $\neg$-move: In the following position $(k-1,s,B,A)$, we have $w \in B \cap A$.

□

For the RE over star-free game, we need a further lemma that gives an easy condition to guarantee that the current sets A and B cannot be separated via a star-free expression. The language we use for the game has words with long strings of the same symbol in them. We call these *a-chains* for $a \in \Sigma$. For example, the word *baabbaaa* has two a-chains of lengths 2 and 3 respectively. We use the GRE game with $s = 0$ to argue about star-free expressions.

Lemma 3.5. *In a position $P = (k, 0, A, B)$ of a game $\mathrm{GRES}(k_0, s_0, A_0, B_0)$, if there are $w \in A$ and $w' \in B$ such that they only differ from each other by lengths of one or more chains of symbols, each of length more than k in both, then D has a winning strategy from position P.*

Proof. We describe a strategy for D. For each move of S, this strategy either wins or maintains the assumptions of the lemma so it is a winning strategy. We consider each possible move of S:

- a-move: S chooses $a \in \Sigma \cup \varepsilon$. Since w has a chain with length more than $k > 0$, clearly $w \neq a$ so D wins.
- $\emptyset$-move: Since $w \in A$, $A \neq \emptyset$ and D wins.
- $\cup$-move: S chooses subsets $A_1, A_2 \subseteq A$. Since $A_1 \cup A_2 = A$, we have $w \in A_i$ for some $i \in \{1,2\}$. D chooses this i and in the following position $(k_i, 0, A_i, B)$ we have $w \in A_i$ and $w' \in B$. In addition, the chains of w and w' that differ are of length more than $k > k_i$. Thus the assumptions still hold.
- cat-move: Let (w_1, w_2) be the split S chooses for $w \in A$ and let $f_{w'} : \mathrm{Sp}^2(w') \to \{1,2\}$ be the function S chooses for $w' \in B$. Let k_1, k_2 be the numbers chosen by S with $k_1 + k_2 + 1 = k$. Since w and w' only differ by the lengths of some chains, for each chain in w we can find the corresponding chain in w'.

  If the split (w_1, w_2) splits no chains where w and w' differ, then we consider the split (w'_1, w'_2) of w' at the corresponding point and in the following position $(k_i, 0, A_i, B_i)$, the assumptions hold since $k_i < k$.

  Now assume (w_1, w_2) splits a chain of length more than k and the length of this chain is different but still more than k in w'. If the length of the chain in w_i is at more than k_i for both i, then we consider a split (w'_1, w'_2) of w' where the same holds. Recall such a split can be found since $k_1 + k_2 + 1 = k$ and the length of the chain is more than k in w' also. Now the assumptions hold in the following position.

  Otherwise, by symmetry we assume that the length of the chain in w_1 is less than or equal to k_1. In this case we consider the split (w'_1, w'_2) of w' where the length of the chain in w'_1 is identical to w_1. Now the lengths of the chains in w_2 and w'_2 are more than k_2 since $k_1 + k_2 + 1 = k$. Thus if the following position is $(k_2, 0, A_2, B_2)$, then the assumptions hold. If the following position is $(k_1, 0, A_1, B_1)$, then either there are still other differing chains of length more than $k > k_1$ and the assumptions hold, or $w_1 = w'_1$ and D has a winning strategy by Lemma 3.4.
- $*$-move: We assume that the star resource $s = 0$ in the position P so S cannot make a $*$-move.
- $\neg$-move: In the following position $(k-1, 0, B, A)$, the assumptions still hold as they are symmetric w.r.t. A and B and $k - 1 < k$.

□

Remark 3.6. The GRE size game can be modified in several ways to obtain different games. The games for RE and RE over star-free are examples of this. Additional operations can be included by adding moves. For example the move corresponding to intersection is the union move with the roles of A and

B switched. One could also have separate resources for different operations or ignore some operations entirely. It is also possible to modify how the resources work with binary moves to track the nesting depth of an operation instead of the number.

4 The succinctness gap between FO and RE

To compare the succinctness of FO and RE, we must restrict the models of FO to *word models*. These are finite models with a linear order and unary predicates to indicate which letter of the alphabet Σ is in each spot. Thus properties of words are often defined in a language of the form $\mathrm{FO}(<, P_1, \ldots, P_n)$.

In his thesis [14] Stockmeyer showed that star-free generalized regular expressions are non-elementarily more succinct than regular expressions. Since there is an elementary translation from FO to star-free expressions [12], this implies that FO is non-elementarily more succinct than RE. The proof of Stockmeyer is quite involved as he encodes computations of Turing machines into star-free expressions. In this section, we show a simple way to obtain the gap between FO and RE via the RE size game. Our proof relies on the following proposition which states that to define a large finite language with a RE, the RE must be quite large as well.

Proposition 4.1. *A finite language L cannot be defined via a* RE *with size less than* $\log|L|$.

Proof. Let L be a finite language and $k_0 < \log|L|$. We consider the game $\mathrm{RES}(k_0, L, \Sigma^* \setminus L)$. We will show that after every move of S, D will either gain a winning strategy via Lemma 3.4, or D can maintain the following two conditions in any position (k, A, B) of the game:

1. $k \leq \log(|A|)$
2. $\Sigma^{>N} := \{w \in \Sigma^* \mid |w| > N\} \subseteq B$ for some $N \in \mathbb{N}$

In the starting position $(k_0, L, \Sigma^* \setminus L)$, we have $k_0 \leq \log(|L|)$ so condition 1 holds. For condition 2, note that since L is finite, $\Sigma^* \setminus L$ includes every word with length greater than the maximum length of words in the language L.

Consider a position (k, A, B) of the game $\mathrm{RES}(k_0, L, \Sigma^* \setminus L)$ and assume conditions 1 and 2 hold. S has five different moves to choose from:

- $*$-move: Since $0 < k \leq \log(|A|)$, we have $|A| \geq 2$ so there is $w \in A$ with $w \neq \varepsilon$. Let $(w_1, w_2, \ldots, w_m)$ be the split chosen by S for w. By condition 2, there is $N \in \mathbb{N}$ such that $\Sigma^{>N} \subseteq B$. Let $v = w_1^{N+1}$. Now $|v| > N$ so $v \in B$. For the split $(w_1, w_1, \ldots, w_1)$ of v S must choose the piece w_1 so in the following position $(k-1, A', B')$, we have $w_1 \in A' \cap B'$ and by Lemma 3.4, D has a winning strategy from this position.
- $\cup$-move: Let $A_1, A_2 \subseteq A$ and $k_1, k_2 < k$ be the choices of S. If either A_i is empty, D chooses the other one and both conditions are trivially maintained. Assume both A_i are non-empty. Since $A_1 \cup A_2 = A$, we obtain $|A_1| + |A_2| \geq |A|$. Now we have $k_i \leq \log(|A_i|)$ for some $i \in \{1, 2\}$, since otherwise

$$\begin{aligned} k &= k_1 + k_2 + 1 > \log(|A_1|) + \log(|A_2|) + 1 \\ &= \log(|A_1||A_2|) + 1 \geq \log(|A_1| + |A_2|) \geq \log(|A|) \geq k, \end{aligned}$$

  which is a contradiction. D chooses such an i, fulfilling condition 1 in the following position is (k_i, A_i, B). Condition 2 is trivially maintained since B remains unchanged in $\cup$-moves.

- cat-move: Let the two possible following positions be $P_i = (k_i, A_i, B_i)$ for $i \in \{1,2\}$. We consider condition 2 first. Let $w \in \Sigma^{>N}$. Let $v \in A$ and let $(v_1, v_2) = v$ be the split chosen by S for v. Now $u = v_1 w \in \Sigma^{>N} \subseteq B$. For the split (v_1, w) of u, if S chooses the piece v_1, then $v_1 \in A_1 \cap B_1$ and by Lemma 3.4, D has a winning strategy from position P_1. Thus we assume that S chooses the piece w and $w \in B_2$. In the same way using the word wv_2, we get $w \in B_1$. Thus, in order to not give D a winning strategy via Lemma 3.4, S must maintain condition 2 for both positions P_i.

  Now let us address condition 1. Since for every $w \in A$ there is $w_1 \in A_1$ and $w_2 \in A_2$ such that $w_1 w_2 = w$, we obtain $|A_1||A_2| \geq |A|$. We again have $k_i \leq \log(|A_i|)$ for some $i \in \{1,2\}$, since otherwise

  $$k = k_1 + k_2 + 1 > \log(|A_1|) + \log(|A_2|) + 1 = \log(|A_1||A_2|) + 1 \geq \log(|A|) \geq k,$$

  which is a contradiction. D again fulfills condition 1 by choosing such an i.
- a- or $\emptyset$-move: Since $0 < k \leq \log(|A|)$, we have $|A| \geq 2$ so $A \not\subseteq \{a\}$ and $A \neq \emptyset$ and D wins the game.

□

The language we use encodes sets of *the cumulative hierarchy*, defined as follows:

$$V_0 := \emptyset$$
$$V_{n+1} := \mathscr{P}(V_n).$$

For each set in the cumulative hierarchy, we define a set of natural encodings. The encodings correspond to the different ways the set could be written down using only set brackets { and }. To differentiate the encoded words from actual set notation, we will use parentheses (and) instead. The encodings are defined as follows:

$$\texttt{enc}(\emptyset) := \{()\}$$
$$\texttt{enc}(X) := \{(e_1 \cdots e_n) \mid e_i \in \texttt{enc}(x_i), x_1 < \cdots < x_n \text{ is a linear order of } X\}.$$

A set has several encodings corresponding to different orders of the elements. For example, the set $V_2 = \{\emptyset, \{\emptyset\}\}$ has the encodings $(()(()))$ and $((())())$.

Let Σ be the alphabet with (and) and let $n \in \mathbb{N}$. We consider the following language:

$$L_n = \bigcup_{X \in V_{n+1}} enc(X).$$

We first define L_n in first-order logic with linear order $<$ and a unary predicate symbol P.

We define some auxiliary formulas. We interpret the predicate P so that the left parentheses satisfy P and the right parentheses do not. We use the formulas $L(x)$ and $R(x)$ to indicate this. We also define the formula $S(x,y)$ that says y is the successor of x.

$$L(x) := P(x), R(x) := \neg P(x), S(x,y) := x < y \wedge \neg \exists z (x < z < y)$$

We will often want to say that the subword from position x_1 to x_2 encodes an instance of a set X. For easy readability of these kinds of statements, we adopt a flexible notation, where capital letters are used as shorthand for pairs of variables, that is to say $X := (x_1, x_2)$. Whenever possible, we shall use only the capital letters but in some cases we need the singular variables also.

We define the formulas $\mathtt{set}_i(X)$ and $X =_i Y$ by mutual recursion. We additionally define formulas $X \in_i Y$, but since these only refer to the formula $\mathtt{set}_i$, they are not essential in the recursion but rather shorthand to make the formulas more readable. The formula $\mathtt{set}_i(X)$ says that X correctly encodes a set in V_i with no repetition. The formula $X \in_i Y$ assumes Y encodes a set and says that X encodes a set in V_i and is an element of the set encoded by Y. Finally, the formula $X =_i Y$ assumes X and Y both encode sets in V_i and says that these sets are the same. The definition by mutual recursion is as follows:

$$\begin{aligned}
\mathtt{set}_0(X) &:= L(x_1) \wedge R(x_2) \wedge S(x_1,x_2) \\
\mathtt{set}_{i+1}(X) &:= x_1 < x_2 \wedge L(x_1) \wedge R(x_2) \\
&\quad \wedge \forall u(x_1 < u < x_2 \to \exists v(x_1 < v < x_2 \wedge (\mathtt{set}_i(u,v) \vee \mathtt{set}_i(v,u)))) \\
&\quad \wedge \forall A \forall B((A \in_i X \wedge B \in_i X \wedge a_1 \neq b_1) \to A \neq_i B)
\end{aligned}$$

$$\begin{aligned}
X \in_i Y &:= y_1 < x_1 < x_2 < y_2 \wedge \mathtt{set}_i(X) \\
&\qquad \wedge \neg \exists U(y_1 < u_1 < x_1 \wedge x_2 < u_2 < y_2 \wedge \mathtt{set}_i(U))
\end{aligned}$$

$$\begin{aligned}
X =_0 Y &:= \top \\
X =_{i+1} Y &:= \forall A(A \in_i X \to \exists B(B \in_i Y \wedge A =_i B)) \\
&\qquad \wedge \forall B(B \in_i Y \to \exists A(A \in_i X \wedge A =_i B))
\end{aligned}$$

We use these auxiliary formulas to define the formula φ_n, which defines the language L_n. The formula φ_n says that the first and last symbol of the word encode a set in V_n with no repetition.

$$\varphi_n := \exists X(\forall z(x_1 \leq z \wedge z \leq x_2) \wedge \mathtt{set}_n(X))$$

From the form of the formulas we see that $\mathrm{sz}(\varphi_n) = \mathcal{O}(c^n)$ for some small constant c.[1]

Now Proposition 4.1 allows us to easily prove a non-elementary succinctness gap between FO and RE. This gap already follows from the work of Stockmeyer [14]. He found a similar gap between star-free expressions and RE and an elementary translation from FO to star-free expressions [12] leads to this result.

Theorem 4.2. $\mathrm{FO}(<,P)$ *is non-elementarily more succinct than* RE *on words.*

Proof. The language L_n is finite and $|L_n| \geq \mathrm{twr}(n)$. We have shown that L_n can be defined in $\mathrm{FO}(<,P)$ via a formula exponential in n. However, if $k < \log(\mathrm{twr}(n)) = \mathrm{twr}(n-1)$, by Theorem 4.1, D has a winning strategy in the game $\mathrm{RES}(k, L, \Sigma^* \setminus L)$. Thus, by Theorem 3.2, there is no RE that defines L with size less than $\mathrm{twr}(n-1)$. □

5 Number of stars in RE over star-free

We shift our attention from the overall size of regular expressions to only the number of stars. Star height famously gives a hierarchy in terms of expressive power for RE [10] and the corresponding result for GRE is a notorious open problem. For the number of stars, a full hierarchy can be trivially obtained already in star height one. On the other hand, for GRE, we have so far been unable to prove results of this nature due to the added complexity brought to the game with full use of complement. We present

[1] Numerical calculations performed with Maple seem to indicate $\mathrm{sz}(\varphi_n) = \mathcal{O}(8^n)$.

an interesting middle ground between RE and GRE we call RE over star-free. For these expressions, star-free, that is FO-definable, properties are combined using the operations of RE. For RE over star-free we show that the number of stars gives a hierarchy in terms of expressive power.

The aforementioned trivial hierarchy for RE is obtained via the expression $a_1^* \cup \cdots \cup a_n^*$ but we omit that proof since we prove the stronger hierarchy for RE over star-free expressions. The language we use is actually definable with n stars already in RE but we show that even if we allow RE over star-free expressions, it still requires n stars to define.

Let $\Sigma_n = \{a_1, \ldots, a_n\}$ be a set of n symbols. We consider the following Σ_n-language:

$$L_n := L\Big(\bigcup_{i \in [n]} (a_1 \cup \cdots \cup a_{i-1} \cup a_i^2 \cup a_{i+1} \cup \cdots \cup a_n)^* \Big)$$

In other words, for each word in $w \in L_n$, there is $i \in [n]$ such that every a_i-chain in w has even length. We don't need the whole language L_n for the game so we use a simple subset instead. For $k \in \mathbb{N}$ and $i \in [n]$, we define

$$L_{n,k} := \{\ell_1, \ldots, \ell_n\} = \{a_1^{2k+1} \cdots a_i^{2k} \cdots a_n^{2k+1} \mid i \in [n]\}.$$

Each ℓ_i is a word that consists of a chain of each symbol a_j in order. The chain of the specific symbol a_i has even length and all other chains of a_j have odd length.

Theorem 5.1. *Any RE over star-free expression R_n with $L(R_n) = L_n$ has at least n stars.*

Proof. Let $n \in \mathbb{N}$ and $k_0 \geq n$. We consider the languages $A_0 := L_{n,k_0}$ and $B_0 := \Sigma_n^* \setminus L_n$. We will show that D has a winning strategy for the game $\mathrm{RES}(k_0, n-1, A_0, B_0)$. Since $A_0 \subseteq L_n$ and $B_0 = \Sigma_n^* \setminus L_n$, D then also has a winning strategy for the game $\mathrm{RES}(k_0, n-1, L_n, \Sigma_n^* \setminus L_n)$. The number k_0 is arbitrary so by Theorem 3.1 the claim follows.

Let (k, s, A, B) be a position in the game $\mathrm{RES}(k_0, n-1, A_0, B_0)$. We will show that D can maintain the following conditions while a $*$-move has not been made. We will also see that if a $*$-move is made while the conditions hold, D gains a winning strategy. The conditions are:

There is $I \subseteq [n]$ such that

1. $|I| > s$,
2. for every $i \in I$ there is $w_i \in A$ and $u_i, v_i \in \Sigma_n^*$ s.t. $\ell_i = u_i w_i v_i$ and $(a_i)^{k+1}$ is a subword of w_i,
3. for every $r \in \Sigma_n^*$ if there are $i, j \in I$ with $u_i r v_j \in B_0$, then $r \in B$.

Intuitively condition 2 says that in the position (k, s, A, B), the set A has some 'descendants' w_i of the original words ℓ_i in A_0. The words u_i and v_i are the parts that have been removed from ℓ_i via cat-moves to obtain w_i. The set I contains the indices that still have descendants in play. Condition 1 states that the number of such indices is always larger than the star resource s. Finally condition 3 says that the set B has versions of the original words in B_0 with some prefix u_i and some suffix v_j removed.

In the starting position $(k_0, n-1, A_0, B_0)$ the conditions hold with $I = [n]$ and for every $i \in I$, $w_i = \ell_i$ and $u_i = v_i = \varepsilon$. We consider each possible move of S and show that in every case either the above conditions are maintained or D wins eventually by a winning strategy described in a previous lemma.

- $\neg$-move: We must first check that while the conditions hold, a $\neg$-move from S leads to a win for D. Let $i \in I$. By condition 2, the word w_i has $(a_i)^{k+1}$ as a subword. Let r be a word obtained from w_i by adding one a_i to this a_i-chain. Since $\ell_i = u_i w_i v_i$ and the a_i-chain in ℓ_i is even, we know the chain in $u_i r v_i$ is odd. The chains of all other a_j are odd in ℓ_i and thus also in $u_i r v_i$ so $u_i r v_i \in B_0$. By

condition 3, we have $r \in B$. If S makes a $\neg$-move, his star resource s becomes 0. In the following position $(k-1,0,B,A)$, we have $r \in B$ and $w_i \in A$ and the two words only differ by the length of a chain with length more than $k-1$ so Lemma 3.5 gives D a winning strategy. This means that while the conditions hold, S can only attempt $\cup$-moves, cat-moves and $*$-moves if he hopes to win.

- $\cup$-move: Let $A_1, A_2 \subseteq A$ be the subsets S chooses. For each $i \in I$, $w_i \in A_1$ or $w_i \in A_2$. Let $I_1, I_2 \subseteq I$ be the sets of indices generated this way. Since $|I| > s$, we have $|I_1| > s_1$ or $|I_2| > s_2$. D chooses the position where this holds. Condition 2 still clearly holds and since B remains unchanged in this move, so does condition 3.

- cat-move: Let $i \in I$ and let $(w_{i,1}, w_{i,2})$ be the split S chooses for w_i. Let $k_1 + k_2 + 1 = k$ and $s_1 + s_2 = s$ be the resource splits of S. Since w_i has $(a_i)^{k+1}$ as a subword, $w_{i,1}$ has $(a_i)^{k_1+1}$ as a subword or $w_{i,2}$ has $(a_i)^{k_2+1}$ as a subword. We divide I into subsets I_1, I_2 according to this condition. Since $|I| > s$, we have $|I_1| > s_1$ or $|I_2| > s_2$. Assume the former. Now condition 2 is satisfied for $w_{i,1}$ by letting $u_{i,1} := u_i$ and $v_{i,1} := w_{i,2}v_i$. For condition 3, let $u_{i,1}rv_{j,1} \in B_0$ for some $r \in \Sigma_n^*$ and $i, j \in I_1$. Now $u_i r w_{j,2} v_j \in B_0$ so by condition 3 in the position before this move, $rw_{j,2} \in B$. For the split $(r, w_{j,2})$ of $rw_{j,2}$ S must choose r to have a chance, since choosing $w_{j,2}$ would result in an identical word on both sides for the position (k_2, s_2, A_2, B_2). So either D has a winning strategy by Lemma 3.4 or $r \in B_1$ for every such r and condition 3 holds for the position (k_1, s_1, A_1, B_1) and D chooses this position. The case of $|I_2| > s_2$ is handled in the same way.

- $*$-move: S can only make this move if $1 \leq s < |I|$ so we have $i, j \in I$ with $i < j$. We will show that this is enough to give D a winning strategy if S makes a $*$-move. Our aim is to show that a word of the form $(w_j)^{m_1}(w_i)^{m_2}$ is in B. We will use condition 3 to show this. Condition 3 requires a word of the form $u_i r v_j$ to be in B_0 and words in B_0 have odd chains of all symbols a_p. Thus we begin by finding odd chains of all symbols in our words.

  Recall that by condition 2, there are $w_i \in A$ and $u_i, v_i \in \Sigma^*$ such that $\ell_i = u_i w_i v_i$ and $(a_i)^{k+1}$ is a subword of w_i. The same holds for j. Let $u \in \{u_i, u_j\}$ be the one of the two words with more odd chains of symbols. If they have the same number of odd chains, we choose, say, the longer word. Choose $v \in \{v_i, v_j\}$ the same way. Next, we will show that for each $p \in [n]$, at least one of the words w_i, w_j, u and v has an odd a_p-chain.

  Recall that the words in A_0 have chains of symbols a_p in order and only the a_i-chain in a word ℓ_i is even while all the others are odd. Furthermore, $\ell_i = u_i w_i v_i$ and w_i has $(a_i)^{k+1}$ as a subword so all chains in u_i are odd except possibly the last. Thus for each odd chain in u_i there is also one of the same symbol in u and the same goes for u_j. Similarly for each odd chain in v_i or v_j there is one in v.

  We now show that for every $p \in [n]$ there is an odd chain in at least one of the words w_i, w_j, u and v. First, let $p < i$. If there is an odd a_p-chain in w_i we are done so let us assume there is not. Now the a_p-chain in w_i is even (possibly empty) and since the chain in $u_i w_i v_i = \ell_i$ is odd, we know the one in u_i is odd. As noted above, an odd chain in u_i means there is also one in u. So in this case there is an odd a_p-chain in w_i or u. The case $p > i$ is very similar and we obtain an a_p-chain in w_i or v. Finally let $p = i$. Now $p < j$ so like above we obtain an odd a_p-chain in w_j or u.

  We now have an odd chain of each a_p among the words w_i, w_j, u and v, but we still need to make sure the specific way we catenate these words does not remove the only odd chains of a symbol by merging them into an even one. Let $f(w)$ be the index of the first symbol of a word w and $l(w)$ the index of the last. By condition 2 we have $f(w_i) \leq i \leq l(w_i)$. The same goes for $f(w_j) \leq j \leq l(w_j)$. We start with $w_j w_i$. By the above we obtain $f(w_i) \leq i < j \leq l(w_j)$ so this catenation cannot result

in any merging of odd chains. Next we add u to the left. If $l(u) = f(w_j)$ and both chains are odd, this merges the chains into an even one. Here we consider two cases. First, if w_j is just an odd a_j-chain, then for some $m_1 \in \{1,2\}$ the a_j-chain in the word $u(w_j)^{m_1}w_i$ is odd. If w_j has other symbols besides a_j, then the word $u(w_j)^2w_i$ has an odd $a_{f(w_j)}$-chain at the start of the second w_j. We have thus obtained $u(w_j)^{m_1}w_i$ with an odd chain of $a_{f(w_j)}$. We finally add v to the right in a similar fashion. If $l(w_i) = f(v)$ and both chains are odd, we again consider the cases of w_i being just an odd a_i-chain or a larger word and we obtain $m_2 \in \{1,2\}$ such that $u(w_j)^{m_1}(w_i)^{m_2}v$ has an odd chain of $a_{l(w_i)}$.

As the words w_i, w_j, u and v have an odd chain of each symbol and we have made sure the catenations did not lose any, our catenated word $u(w_j)^{m_1}(w_i)^{m_2}v$ is now in B_0. Since $u \in \{u_i, u_j\}$ and $v \in \{v_i, v_j\}$, by condition 3, $(w_j)^{m_1}(w_i)^{m_2} \in B$.

Let us finish by showing how this gives D a winning strategy after the $*$-move in progress. S must give splits for w_i and w_j and every piece of these splits is in the left set of the following position, A'. S must also choose a piece of every split of $(w_j)^{m_1}(w_i)^{m_2}$ to add to the right set, B'. The split of $(w_j)^{m_1}(w_i)^{m_2}$ we are interested in is the one where each subword w_i and w_j is split according to the splits given by S for w_i and w_j. For this split, S must choose one of the pieces already in A' to also be in B'. Thus, in the following position $(k-1, s-1, A', B')$, there is an identical word on both sides and D has a winning strategy by Lemma 3.4. Thus if S makes a $*$-move while the conditions hold, D eventually wins.

□

6 Conclusion

We have presented a formula size game for GRE, RE and a middle ground between these we call RE over star-free expressions. We used the RE version to reprove a non-elementary succinctness gap between FO and RE via a large finite language. For RE over star-free we showed that the number of stars gives a full hierarchy in terms of expressive power. As the astute reader has noted, we have not used the full GRE size game in this paper. This is due to the considerable combinatorial complexity of the game. A clear goal for further research is to find some handle on this complexity at least for some problems. A good first candidate is to prove that there is a star height one language that requires two stars to define via a GRE.

As noted in Remark 3.6, the games can be modified to isolate different operations with different resources or counting the nesting depth of some operations instead of the number. This means that the games could naturally be used to investigate any problem having to do with bounds on operators such as the generalized star height problem.

References

[1] M. Adler & N. Immerman (2003): *An* n! *lower bound on formula size*. *ACM Trans. Comput. Log.* 4(3), pp. 296–314, doi:10.1145/772062.772064.

[2] H-D. Ebbinghaus & J. Flum (1995): *Finite Model Theory: First Edition*. Springer Berlin Heidelberg, Berlin, Heidelberg, doi:10.1007/978-3-662-03182-7.

[3] L. C. Eggan (1963): *Transition graphs and the star-height of regular events. Michigan Math. J.* 10(4), pp. 385–397, doi:10.1307/mmj/1028998975.

[4] A. Ehrenfeucht & P. Zeiger (1974): *Complexity Measures for Regular Expressions*. In: *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing*, STOC '74, Association for Computing Machinery, New York, NY, USA, p. 75–79, doi:10.1145/800119.803886.

[5] K. Ellul, B. Krawetz, J. Shallit & M. Wang (2005): *Regular Expressions: New Results and Open Problems*. *J. Autom. Lang. Comb.* 10(4), p. 407–437, doi:10.25596/jalc-2005-407.

[6] W. Gelade (2010): *Succinctness of regular expressions with interleaving, intersection and counting*. *Theor. Comput. Sci.* 411(31-33), pp. 2987–2998, doi:10.1016/j.tcs.2010.04.036.

[7] W. Gelade & F. Neven (2012): *Succinctness of the Complement and Intersection of Regular Expressions*. *ACM Trans. Comput. Logic* 13(1), doi:10.1145/2071368.2071372.

[8] H. Gruber & M. Holzer (2008): *Finite Automata, Digraph Connectivity, and Regular Expression Size*. In L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir & I. Walukiewicz, editors: *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings*, *Lecture Notes in Computer Science* 5126, Springer, pp. 39–50, doi:10.1007/978-3-540-70583-3_4.

[9] H. Gruber & M. Holzer (2009): *Tight Bounds on the Descriptional Complexity of Regular Expressions*. In V. Diekert & D. Nowotka, editors: *Developments in Language Theory*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 276–287, doi:10.1007/978-3-642-02737-6_22.

[10] K. Hashiguchi (1988): *Algorithms for Determining Relative Star Height and Star Height*. *Inf. Comput.* 78(2), pp. 124–169, doi:10.1016/0890-5401(88)90033-8.

[11] J. E. Hopcroft, R. Motwani & J. D. Ullman (2006): *Introduction to Automata Theory, Languages, and Computation*, 3rd edition. Addison-Wesley Longman Publishing Co., Inc., USA.

[12] R. McNaughton & S. A. Papert (1971): *Counter-Free Automata (M.I.T. Research Monograph No. 65)*. The MIT Press.

[13] A. A. Razborov (1990): *Applications of matrix methods to the theory of lower bounds in computational complexity*. *Combinatorica* 10(1), pp. 81–93, doi:10.1007/BF02122698.

[14] L. Stockmeyer (1974): *The complexity of decision problems in automata theory and logic*. Ph.D. thesis, Massachusetts Institute of Technology.

[15] Q. Yan (2007): *Classifying regular languages by a split game*. *Theoretical Computer Science* 374(1), pp. 181 – 190, doi:10.1016/j.tcs.2006.12.041.