On is an n-MCFL

Kilian Gebhardt, Frédéric Meunier, Sylvain Salvati

O_n IS AN n-MCFL

KILIAN GEBHARDT[1], FRÉDÉRIC MEUNIER[2], AND SYLVAIN SALVATI[3]

Abstract. Commutative properties in formal languages pose problems at the frontier of computer science, computational linguistics and computational group theory. A prominent problem of this kind is the position of the language O_n, the language that contains the same number of letters a_i and $\bar{a}_i$ with $1 \leqslant i \leqslant n$, in the known classes of formal languages. It has recently been shown that O_n is a Multiple Context-Free Language (MCFL). However the more precise conjecture of Nederhof that O_n is an MCFL of dimension n was left open. We present two proofs of this conjecture, both relying on tools from algebraic topology. On our way, we prove a variant of the necklace splitting theorem.

1. Introduction

The language O_n is the language built on the alphabet $\Sigma_n = \{a_i, \bar{a}_i \mid i \in [n]\}$ and that contains exactly all words w which, for every i in $[n]$, have the same number of occurrences of a_i and $\bar{a}_i$. Writing $|w|_c$ the number of occurrences of the letter c in w, this condition becomes $|w|_{a_i} = |w|_{\bar{a}_i}$ for all i in $[n]$. The problem of situating the languages O_n within known classes of languages has been raised at least in two different communities.

The first one is that of computational linguistics. This problem has attracted attention with a language called MIX (also called the Bach language as it was introduced by Bach [Bac81, Bac88, Pul83]) that is rationally equivalent to O_2. It was related to the research program of Joshi [Jos85] which consists in formally describing the class of languages corresponding to Natural Languages. According to Joshi's terminology, this is the class of Mildly Context Sensitive Languages. This program tries to give abstract properties of this class while describing possible candidates [Wei88, JSW91]. Among such candidates, a powerful one is formed by Multiple Context Free Languages (MCFL) [SMFK91]. According to Joshi *et al.* [JSW91] the language "MIX can be regarded as the extreme case of free word order" and Mildly Context Sensitive Languages should "perhaps" not contain MIX (and thus O_2). This paper also mentions that the position of MIX in classes of languages, such as Tree Adjoining Languages, is not known and difficult to establish. In particular, it stresses that it is not known whether MIX is an Indexed Language.

The second community that has also shown interest for the problem is that of computational group theory which tries to identify properties of groups by means of properties of their word problem. The word problem for a group consists in describing the language of words that are equal to zero for a given presentation (all presentations giving rise to rationally equivalent languages). For example, Muller and Schuppe have characterized virtually free groups as exactly those groups whose word problems are solved by context free grammars [MS83]. A question that has been raised by that community is whether O_2—which coincides with the

[1]Technische Universität Dresden, Germany, kilian@gebhardt.xyz.
[2]CERMICS, École des Ponts ParisTech, France, frederic.meunier@enpc.fr.
[3]Université de Lille, INRIA, CRIStAL, France, sylvain.salvati@univ-lille.fr.

language corresponding to the word problem for the additive group $(\mathbb{Z}^2, +)$—is an Indexed Language [Gil05]. This question remains open. MCFL form a natural generalization of "copyless" Macro-Languages (see [SK08] for a discussion) and Macro-Languages are equivalent to Indexed Languages. This makes the question of whether O_2 belongs to MCFL relevant.

A first important result in that line of research is that O_2 is not a well-nested MCFL of dimension 2 [KS12], which solves a long-standing open problem raised by Joshi [Jos85]. Subsequently it has been shown that it is actually an MCFL [Sal15] and more precisely an MCFL of dimension 2 (a 2-MCFL). Nederhof [Ned16] has given a similar proof of the same result. Later he conjectured that O_n is an n-MCFL [Ned17]. As pointed in this work, a simple pumping argument shows that O_n cannot be an MCFL of dimension strictly smaller than n. More recently, a breakthrough has been achieved by Ho [Ho18] who proved that O_n is an MCFL for every n. However the MCFL constructed in the proof is of dimension larger than n, namely $8\left\lfloor\frac{n+1}{2}\right\rfloor - 2$. All proofs related to these results are based on algebraic topology. While the proofs of [Sal15] and [Ned16] strongly rely on topological properties of the plane (existence of winding numbers of curves around points), the aforementioned proof by Ho is based on the well-known Borsuk–Ulam theorem, a powerful theorem from algebraic topology which holds in any dimension. More specifically, it uses a combinatorial application of this theorem, due to Alon and West [AW86]: the necklace splitting theorem. In this paper, we also rely on related tools to prove Nederhof's conjecture, namely that O_n is an n-MCFL. Nederhof actually conjectures that a particular Multiple Context-Free Grammar (MCFG) of dimension n defines O_n. We prove a slightly stronger result by showing that a grammar that uses a more restricted kind of rules is sufficient to define O_n.

The article is structured as follows: In Section 2, we introduce notation regarding formal languages and MCFG as well as the grammar G_n, which is an MCFG of dimension n. Section 3 establishes the main result namely that the language of G_n is O_n using a decomposition lemma. The decomposition lemma can on the one hand be derived from a variant of the necklace splitting theorem that we prove in Section 4. Alternatively, it can be obtained via purely combinatorial proofs presented in Section 5.

2. Background on Multiple Context Free Grammars

We write $[n]$ for the set $\{1, \dots, n\}$. For a given finite set Σ, also called *alphabet*, we write Σ^* for the monoid freely generated by Σ, and Σ^+ for the free semigroup generated by Σ. The elements of Σ are called *letters* while the elements of Σ^* and Σ^+ are called *strings* or *words* and we write ε for the empty word. Given a word w, we write $|w|$ for its length, and $|w|_c$ for the number of occurrences of the letter c in w. A *language* is a subset of Σ^*.

We define the language O_n as $\{w \in \Sigma_n^* \mid |w|_{a_i} = |w|_{\bar{a}_i} \text{ for } i \in [n]\}$ where Σ_n is the alphabet $\{a_i, \bar{a}_i \mid i \in [n]\}$. For α in Σ_n, writing $\bar{\alpha}$ exchanges a_i and $\bar{a}_i$: if α is a_i, then $\bar{\alpha}$ is $\bar{a}_i$; if α is $\bar{a}_i$, then $\bar{\alpha}$ is a_i. Two letters α and β of Σ_n are *compatible* when $\alpha = \bar{\beta}$. We extend the $(\bar{\cdot})$ operation to words in Σ_n^* as follows: the word $\bar{w}$ is obtained from w by applying $(\bar{\cdot})$ to each of its letters.

A ranked alphabet Ω is a pair $(\mathcal{A}, \rho)$ where $\mathcal{A}$ is a finite set and ρ is a function from $\mathcal{A}$ to $\mathbb{N}$. For a in $\mathcal{A}$, the integer $\rho(a)$ is the *rank* of a. We shall write $\Omega^{(n)}$ for the set $\{a \in \mathcal{A} \mid \rho(a) = n\}$. The *dimension* of a ranked alphabet is the maximal rank of its elements.

A Multiple Context Free Grammar (MCFG) G is a tuple (Ω, Σ, R, S) where Ω is a ranked alphabet of *non-terminals*, Σ is a finite set of *letters*, R is a set of *rules* and S is an element

of $\Omega^{(1)}$ called *initial non-terminal*. The rules in R are of the form

$$A(w_1, \ldots, w_n) \Rightarrow B_1(x_{1,1}, \ldots, x_{1,l_1}), \ldots, B_p(x_{p,1}, \ldots, x_{p,l_p}) \tag{1}$$

where A is in $\Omega^{(n)}$, B_k is in $\Omega^{(l_k)}$, the $x_{k,j}$ are pairwise distinct variables and the w_j are elements of $(\Sigma \cup X)^*$ with $X = \{x_{k,j} \mid k \in [p] \wedge j \in [l_k]\}$ and with the restriction that each $x_{k,j}$ may have at most one occurrence in the string $w_1 \cdots w_n$. Note that p may be equal to 0, in which case the right part of the rule (the one on the right of the $\Rightarrow$ symbol) is empty. Then we may write the rule by omitting the symbol $\Rightarrow$. The *dimension* of an MCFG is that of its ranked alphabet of non-terminals. An MCFG of dimension at most n is an n-MCFG.

An MCFG such as G defines *judgments* of the form $\vdash_G A(s_1, \ldots, s_n)$ where A is in $\Omega^{(n)}$ and the s_j belong to Σ^*. The notion of derivable judgments is defined inductively: suppose we are given p derivable judgments $\vdash_G B_k(s_{k,1}, \ldots, s_{k,l_k})$ where $B_k \in \Omega^{(l_k)}$ for k in $[p]$. For each rule of the form (1), the judgment $\vdash_G A(s_1, \ldots, s_n)$ is *derivable* when each s_j is obtained from w_j by replacing each occurrence of the variable $x_{k,j}$ by $s_{k,j}$. The language defined by G, denoted by $\mathcal{L}(G)$, is the set $\{w \in \Sigma^* \mid \vdash_G S(w) \text{ is derivable}\}$. The class of languages that are definable by MCFGs is the class of *Multiple Context-Free Languages* (MCFL). Likewise, the class of languages definable by n-MCFGs is the class of *n-Multiple Context-Free Languages* (n-MCFL).

We define now G_n, the central n-MCFG for which we prove that it generates O_n. It uses two non-terminals S and I that are respectively of rank 1 and n. The non-terminal S is the initial one. The alphabet of G_n is Σ_n. The rules of the grammar G_n are the following:

(1) $S(x_1 \cdots x_n) \Rightarrow I(x_1, \ldots, x_n)$.
(2) $I(w_1, \ldots, w_n) \Rightarrow I(x_1, \ldots, x_n), I(y_1, \ldots, y_n)$
for all $w_1, \ldots, w_n \in \{x_1, \ldots, x_n, y_1, \ldots, y_n\}^*$ such that $w_1 \cdots w_n = x_1 y_1 \cdots x_n y_n$.
(3) $I(w_1, \ldots, w_n) \Rightarrow I(x_1, \ldots, x_n)$ for all $w_1, \ldots, w_n$ and all $\alpha \in \Sigma_n$, $k, \ell \in [n]$ such that $w_j = x_j$ for $j \neq k, \ell$ and
- $k \neq \ell$ implies $w_k \in \{\alpha x_k, x_k \alpha\}$ and $w_\ell \in \{\bar{\alpha} x_\ell, x_\ell \bar{\alpha}\}$.
- $k = \ell$ implies $w_k = \alpha x_k \bar{\alpha}$.

(4) $I(\varepsilon, \ldots, \varepsilon)$.

Items numbered (2) and (3) describe finite sets of rules. The rules (2) are parametrized by a particular factorization $(w_1, \ldots, w_n)$ of $x_1 y_1 \cdots x_n y_n$. For example, when $n = 3$ letting $w_1 = x_1 y_1 x_2$, $w_2 = y_2 x_3$ and $w_3 = y_3$ is such a factorization; we have $w_1 w_2 w_3 = x_1 y_1 x_2 y_2 x_3 y_3$. A rule of the form (3) adds a compatible pair of letters at distinct endpoints of the words. For example, $I(a_1 x_1, x_2, \bar{a}_1 x_3, x_4) \Rightarrow I(x_1, x_2, x_3, x_4)$ is such a rule for $n = 4$.

Example 1. The grammar G_2 contains the following rules:

$$\begin{aligned}
S(x_1 x_2) &\Rightarrow I(x_1, x_2) \\
I(x_1 y_1 x_2 y_2, \varepsilon) &\Rightarrow I(x_1, x_2),\ I(y_1, y_2) \\
I(x_1 y_1 x_2, y_2) &\Rightarrow I(x_1, x_2),\ I(y_1, y_2) \\
I(x_1 y_1, x_2 y_2) &\Rightarrow I(x_1, x_2),\ I(y_1, y_2) \\
I(x_1, y_1 x_2 y_2) &\Rightarrow I(x_1, x_2),\ I(y_1, y_2) \\
I(\varepsilon, x_1 y_1 x_2 y_2) &\Rightarrow I(x_1, x_2),\ I(y_1, y_2) \\
I(\alpha x_1 \bar{\alpha}, x_2) &\Rightarrow I(x_1, x_2) \quad \alpha \in \Sigma_2 \\
I(\alpha x_1, \bar{\alpha} x_2) &\Rightarrow I(x_1, x_2) \quad \alpha \in \Sigma_2
\end{aligned}$$

$$
\begin{array}{rcll}
I(\alpha x_1, x_2\bar{\alpha}) & \Rightarrow & I(x_1, x_2) & \alpha \in \Sigma_2 \\
I(x_1\alpha, \bar{\alpha}x_2) & \Rightarrow & I(x_1, x_2) & \alpha \in \Sigma_2 \\
I(x_1\alpha, x_2\bar{\alpha}) & \Rightarrow & I(x_1, x_2) & \alpha \in \Sigma_2 \\
I(x_1, \alpha x_2\bar{\alpha}) & \Rightarrow & I(x_1, x_2) & \alpha \in \Sigma_2 \\
I(\varepsilon, \varepsilon) & & &
\end{array}
$$

It is usual to present derivations as trees where nodes have the following form:

$$\frac{\vdash_G B_1(s_{1,1}, \ldots, s_{1,l_1}) \cdots \vdash_G B_p(s_{p,1}, \ldots, s_{p,l_p})}{\vdash_G A(s_1, \ldots, s_n)}$$

when from the derivations of the $\vdash_G B_i(s_{i,1}, \ldots, s_{i,l_i})$ we can derive $\vdash_G A(s_1, \ldots, s_n)$ using a rule of G. When p equals 0, there is no derivation above the bar. This means that rules with no right-hand part form the leaves of these trees.

Using this notation we can show as follows that $a_1a_1\bar{a}_2\bar{a}_1\bar{a}_1a_2$ is in the language of G_2. We use colors to identify each letter and allow one to infer the rules used in the derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash_{G_2} I(\varepsilon,\varepsilon)}}{\vdash_{G_2} I(a_1,\bar{a}_1)} \quad \dfrac{\dfrac{\dfrac{}{\vdash_{G_2} I(\varepsilon,\varepsilon)}}{\vdash_{G_2} I(\bar{a}_2,a_2)}}{\vdash_{G_2} I(a_1\bar{a}_2,\bar{a}_1a_2)}}{\vdash_{G_2} I(a_1,a_1\bar{a}_2\bar{a}_1\bar{a}_1a_2)}}{\vdash_{G_2} S(a_1a_1\bar{a}_2\bar{a}_1\bar{a}_1a_2)}$$

3. Main result

Our main theorem is that the language of G_n is O_n. The inclusion of $\mathcal{L}(G_n)$ into O_n is obvious and the challenge consists in proving the converse inclusion. We prove actually a stronger statement: $\vdash_{G_n} I(s_1, \ldots, s_n)$ *is derivable for every* $s_1 \cdots s_n$ *in* O_n. With this statement, the desired inclusion is immediate: for $w \in O_n$, set $s_1 = w$ and $s_j = \varepsilon$ for all $j \geqslant 2$ and apply rule (1).

A tuple $(s_1, \ldots, s_n)$ is *reducible* if there are at least two compatible letters among the endpoints of the s_j's; otherwise the tuple is *irreducible*. Repeated applications of rules of the form (3) allow to derive reducible tuples from irreducible ones. Irreducible tuples are dealt with the following "decomposition lemma," which shows how a rule of the form (2) is applied to derive an irreducible tuple from smaller elements in O_n. While rule (4) provides the base case for the induction, this lemma is the main technical result towards the proof of Theorem 1. It relies on Theorem 2, a result stated and proved in Section 4, and formulated with the terminology of the necklace splitting theorem.

Lemma 1 (Decomposition lemma)**.** *Consider an irreducible tuple* $(s_1, \ldots, s_n)$ *in* $(\Sigma_n^+)^n$*, where each* s_j *is of length at least* 2*. If* $s_1 \cdots s_n$ *belongs to* O_n*, then there exist two tuples* $(u_1, u_3, \ldots, u_{2n-1})$ *and* $(u_2, u_4, \ldots, u_{2n})$ *in* $(\Sigma_n^*)^n$ *and integers* $0 = k_0 \leqslant k_1 \leqslant \cdots \leqslant k_n = 2n$ *such that*

(I) $u_1u_3 \cdots u_{2n-1}$ *and* $u_2u_4 \cdots u_{2n}$ *are both nonempty elements of* O_n*, and*
(II) $s_j = u_{k_{j-1}+1}\, u_{k_{j-1}+2} \cdots u_{k_j}$ *for each* $j \in [n]$.

In particular, $s_1 \cdots s_n = u_1 u_2 \cdots u_{2n}$.

Proof. We distinguish the cases $n = 1$ and $n \geqslant 2$.

We deal first with the case $n = 1$. In that case, $\Sigma_n = \{a_1, \bar{a}_1\}$. As s_1 is irreducible, without loss of generality, we may assume that $s_1 = a_1 w a_1$. We consider prefixes u' of w of increasing length, from $|u'| = 0$ to $|u'| = |w|$. Since s_1 belongs to O_1, the quantity $|a_1 u'|_{a_1} - |a_1 u'|_{\bar{a}_1}$ starts with the value 1 and finishes with the value -1, and changes by steps of one unit. There is therefore a prefix u' of w such that $a_1 u'$ belongs to O_1. Setting $u_1 = a_1 u'$ and $u_2 \in \Sigma_1^+$ such that $s_1 = u_1 u_2$ makes the job.

We deal now with the case $n \geqslant 2$. The proof of Theorem 2 builds explicitly u_ℓ's satisfying the desired properties: property (I) is a consequence of the fact that each of A and B are balanced; property (II) is a consequence of the fact that the endpoints of the s_j's form cuts. □

From this the inclusion follows easily.

Proposition 1. *The judgment* $\vdash_{G_n} I(s_1, \ldots, s_n)$ *is derivable for every* $s_1 \cdots s_n$ *in* O_n. *In particular, we have* $O_n \subseteq \mathcal{L}(G_n)$.

Proof. As mentioned above, the second part of the statement is a direct consequence of the first part. We proceed by induction on the pairs $(|s_1 \cdots s_n|, e)$ ordered lexicographically, where e is the number of s_j equal to ε.

Suppose first that $(s_1, \ldots, s_n)$ is reducible. A rule of the form (3) shows that we can derive the judgment from another judgment $\vdash_{G_n} I(s'_1, \ldots, s'_n)$, with $|s'_1 \cdots s'_n| < |s_1 \cdots s_n|$. The induction hypothesis provides the conclusion.

Suppose now that $(s_1, \ldots, s_n)$ is irreducible. Four cases are in order, distinguished according to the possible lengths of the s_j's.

The first case is when at least one s_j is of length 1. Without loss of generality we may assume that $s_j = a_1$. As $s_1 \cdots s_n$ is in O_n, there is a k such that $s_k = v_1 \bar{a}_1 v_2$ for some v_1 and v_2 in Σ_n^*. The other case being similar, we suppose that $j < k$. Define $t_j = a_1$, $t_k = \bar{a}_1$, and $t_\ell = \varepsilon$ for $\ell \neq j, k$; define also $(u_1, \ldots, u_n) = (s_1, \ldots, s_{j-1}, s_{j+1}, \ldots, s_{k-1}, v_1, v_2, s_{k+1}, \ldots, s_n)$. Using rule (4) and a rule of the form (3), we get that $\vdash_{G_n} I(t_1, \ldots, t_n)$ is derivable. The judgment $\vdash_{G_n} I(u_1, \ldots, u_n)$ is derivable by induction. Then using a rule of the form (2) shows that $\vdash_{G_n} I(s_1, \ldots, s_n)$ is derivable. More precisely, we instantiate each variable x_ℓ with t_ℓ and each variable y_ℓ with u_ℓ in the following rule

$$I(x_1 y_1, \ldots, x_{j-1} y_{j-1}, x_j, y_j, \ldots, x_{k-2} y_{k-2} x_{k-1}, y_{k-1} x_k y_k, \ldots, x_n y_n) \Rightarrow I(x_1, \ldots, x_n), I(y_1, \ldots, y_n) \ .$$

The second case is when all s_j are equal to ε. The conclusion follows from an application of rule (4).

The third case is when some s_j but not all are equal to ε and no s_j is of length 1. There is then an $j \in [n-1]$ such that either $s_j = \varepsilon$ and $|s_{j+1}| > 1$, or $|s_j| > 1$ and $s_{j+1} = \varepsilon$. By symmetry, we suppose that $s_j = \varepsilon$ and $|s_{j+1}| > 1$. As $|s_{j+1}| > 1$, we have that $s_{j+1} = v_1 v_2$ with v_1 and v_2 in Σ^+. Define $(s'_1, \ldots, s'_n) = (s_1, \ldots, s_{j-1}, v_1, v_2, s_{j+2}, \ldots, s_n)$. The judgment $\vdash_{G_n} I(s'_1, \ldots, s'_n)$ is derivable by induction (we have a smaller e). The judgment $\vdash_{G_n} I(\varepsilon, \ldots, \varepsilon)$ is derivable from an application of rule (4). Then using a rule of the form (2) shows that $\vdash_{G_n} I(s_1, \ldots, s_n)$ is derivable. More precisely, we instantiate each variable x_ℓ with s'_ℓ and

each variable y_ℓ with ε in the following rule:

$$I(x_1y_1, \ldots, x_{j-1}y_{j-1}, \varepsilon, x_iy_jx_{j+1}y_{j+1}, \ldots, x_ny_n) \Rightarrow I(x_1, \ldots, x_n), I(y_1, \ldots, y_n) \ .$$

The fourth case satisfies the conditions of Lemma 1 ("decomposition lemma"), which shows that $\vdash_{G_n} I(s_1, \ldots, s_n)$ is derivable by an application of a rule of the form (2) and by induction. □

From this we can derive our main theorem.

Theorem 1. *The language O_n is an n-MCFL.*

Remark. Theorem 2 actually implies a version of Lemma 1 that also holds for reducible tuples albeit only if n is at least 2 and if we permit to decompose $(s_1, \ldots, s_n)$ in more versatile tuples. This corresponds to a slightly more liberal definition of the rules (2). More precisely, we may add to the rules (2) for each $j \in [n]$ the rule:

$$(*) \qquad I(x_1y_1, \ldots, x_{j-1}y_{j-1}, \boldsymbol{y_jx_j}, x_{j+1}y_{j+1}, \ldots, x_ny_n) \Rightarrow I(x_1, \ldots, x_n),\ I(y_1, \ldots, y_n) \ .$$

We have chosen to exclude rules $(*)$ to emphasize the surprising simplicity of the rules (2) which allow to decompose every irreducible tuple. Moreover, we want to contrast the grammar we obtain with the one that Nederhof [Ned17] conjectures to capture O_n. Nederhof proposes binary rules of the following form:

$$A(w_1, \ldots, w_n) \Rightarrow A(x_1, \ldots, x_n),\ A(y_1, \ldots, y_n)$$

where $w_1 \cdots w_n$ is obtained by shuffling the words $x_1 \cdots x_n$ and $y_1 \cdots y_n$, i.e., $|w_1 \cdots w_n| = 2n$, removing all occurrences of y_j's from $w_1 \cdots w_n$ yields $x_1 \cdots x_n$ and, analogously, removing all occurrences of x_j's from $w_1 \cdots w_n$ yields $y_1 \cdots y_n$. Furthermore, a w_k may not contain an occurrence of x_jx_{j+1} or of y_jy_{j+1} for some $j \in [n-1]$. The rules (2) may be obtained from Nederhof's by forbidding the occurrence of x_jx_{j+1} and of y_jy_{j+1} not only in the w_k's but rather in the combined word $w_1 \cdots w_n$. Note however that this additional restriction implies that rules (3) need to allow the removal of compatible letters at arbitrary endpoints of the words of a tuple. The corresponding rules of Nederhof's grammar are less liberal. Were we to remove rules (3) and treat the elimination of compatible letters as in [Ned17], then we would need to add the rules $(*)$. Clearly, these rules form a strict subset of those proposed by Nederhof as w_k's are of length 2 and at most one occurrence of x_jx_{j+1} is allowed in $w_1 \cdots w_n$. In this sense, our grammar is simpler than Nederhof's. Notably, also Nederhof [Ned17, Sec. 5.3] conjectures for O_3 that some of the rules of his grammar are redundant.

4. Necklace splitting and proof of the decomposition lemma

In this section, we prove a combinatorial theorem in the tradition of the necklace splitting problem. The theorem almost implies Lemma 1, but while it does not require irreducibility, it misses the property (II). Its proof however gives the construction for the case $n > 1$ of Lemma 1.

This combinatorial theorem is formulated without the terminology of languages and grammars so as to make it understandable easily without background in this area. For readers who are more familiar to manipulating words, the vocabulary of the necklace problem translates easily to that of words: necklaces become words and beads become letters.

In the traditional version of the necklace splitting theorem, there is an open necklace with beads of n different types, and an even number of beads of each type. The necklace splitting

theorem ensures that such a necklace can always be split between two thieves with no more than n cuts so that each thief gets the same amount of each type. The cuts have to leave the beads untouched. Here, we keep the same setting, except that a bead can be either "positive" or "negative." The *amount* of a type in a necklace or in a collection of necklaces is the number of positive beads of this type minus the number of negative beads of this type. A necklace or a collection of necklaces is *balanced* if the amount of each type is zero.

Theorem 2. *Consider a collection of n open necklaces with positive and negative beads. Suppose that there are $n \geqslant 2$ types of beads and that each of the necklaces has at least two beads. If the collection is balanced, then there is a way to cut the necklaces using at most n cuts in total and partition the subnecklaces into two parts so that each part is balanced, gets at least one bead (and thus at least two), and is formed by at most n subnecklaces.*

The connection to Lemma 1 is as follows:

- The n types of beads are the numbers $1, \ldots, n$, the letter a_i representing a *positive* bead of type i and the letter $\bar{a}_i$ representing a *negative* bead of type i.
- The n open necklaces correspond to the words $s_1, \ldots, s_n$.
- The two parts of the at most n subnecklaces are the tuples $(u_1, \ldots, u_{2n-1})$ and $(u_2, \ldots, u_{2n})$ in the lemma, which, as in the theorem, are required to be balanced and to contain at least one letter each.

The proof of Theorem 2 relies on the following proposition, which is actually the traditional necklace splitting theorem extended to negative beads (the relaxation of the parity condition of the number of beads is standard; see, e.g., [AMS06, Section 5.1]).

Proposition 2. *Consider a necklace with positive and negative beads. Suppose that there are $n \geqslant 1$ types of beads. Then the necklace can be split between two thieves with no more than n cuts so that the amount of beads received by the thieves differ by at most one unit for each type. It is moreover possible to choose which thief receives an extra bead for each type which requires so.*

The types requiring that one of the thieves receives an extra bead are precisely those whose total amount of beads is an odd number.

Proof of Proposition 2. We proceed in two steps. In a first step, we prove a continuous version, in which we are allowed momentarily to locate cuts on beads themselves. In a second step, we show how to move the cuts so as to get a splitting with cuts leaving the beads untouched.[1]

We identify the necklace with $[0, 1]$ and the beads with intervals included in $[0, 1]$, all of same length (this length is the inverse of the total number of beads in the necklace), and with disjoint interiors. Assume that these small intervals are all open. (This assumption is made to ease the proof, but actually whether these small intervals contain or not their boundaries does not matter.) Define

$$g_i(x) \longmapsto \begin{cases} +1 & \text{if } x \text{ is in a interval corresponding to a positive bead of type } i. \\ -1 & \text{if } x \text{ is in a interval corresponding to a negative bead of type } i. \\ 0 & \text{otherwise.} \end{cases}$$

[1] An alternative purely combinatorial proof is presented in Section 5.2.

According to the Hobby–Rice theorem [HR65], there are points $0 = x_0 < x_1 < \cdots < x_r < x_{r+1} = 1$ with $r \leqslant n$ such that for all $i \in [n]$

$$\sum_{j=1}^{r+1} (-1)^j \int_{x_{j-1}}^{x_j} g_i(u) \,\mathrm{d}\, u = 0 \ .$$

(Here, we take the formulation given by Pinkus [Pin76].) The r points $x_1, \ldots, x_r$ can be interpreted as cuts. The intervals (x_{j-1}, x_j) with j odd are given to one thief, and the intervals (x_{j-1}, x_j) with j even are given to the other thief. Each thief gets the same amount of each type. The only problem is that some cuts may be located on beads.

The second step of the proof consists in explaining how to move these cuts so that none of them touch the beads anymore, without creating a difference of more than one unit between the amounts received by the thieves for each type. We can make that each bead is cut at most twice since moving two cuts inside a bead in the same direction and by the same distance does not change the amounts received by the thieves. If a bead is cut twice, we can similarly move the two cuts until one of them is located between two beads. (Doing this, we can have several cuts located at the same position between two beads, but this is not an issue.) So, we can assume that each bead is cut at most once and that the two parts of a cut bead go to distinct thieves. If a type has two beads or more touched by a cut, then we can move two cuts so that one at least reaches a position between two beads, without changing the amounts received by each thief. Thus, we can assume that each type is cut at most once. We finish the proof by noting that if a type has a bead that is cut, it means that each thief received a non-integral amount of the corresponding type, i.e., a half-integer, and moving the cut arbitrarily leads to the desired splitting. □

Proof of Theorem 2. Denote by $s_1, \ldots, s_n$ the n necklaces. We assume that there are no two beads located at the endpoints of some necklaces and that are of the same type but of opposite signs, for otherwise there would be an easy solution: cut these two beads from the necklaces, form a balanced part with them, and leave the rest of the remaining beads for the second balanced part. The number of cuts would then be $2 \leqslant n$ (or, if these two beads formed a necklace of their own, the solution would need no cut), and the number of subnecklaces in the parts would be 2 and n (or 1 and $n-1$). Making this assumption corresponds to considering only the irreducible case in the terminology of Theorem 1.

A natural idea would be to apply a result like Proposition 2 to the "big" necklace $s = s_1 \cdots s_n$ obtained by appending the necklaces in their index order. The first issue with this idea is that one thief might get nothing. This can easily be dealt with as done below. The second issue is that, even though there are $2n$ subnecklaces in total, one thief might get more than n subnecklaces.[2] Instead we consider another big necklace, which we will denote by s' and which we define now.

We start by defining s'_1 to be s_1 from which the left-most bead has been removed, and s'_n to be s_n from which the right-most bead has been removed. Note that without loss of generality, we can assume that the left-most bead of s_1 is a positive bead of type 1 and that the right-most bead of s_n is a positive bead of type 1 or n. We thus consider two cases:

(i) The left-most bead of s_1 and the right-most bead of s_n are both positive beads of type 1.

[2] In essence, this is the reason why Ho [Ho18] could only show that O_n is an $\left(8\left\lfloor\frac{n+1}{2}\right\rfloor - 2\right)$-MCFL.

(ii) The left-most bead of s_1 is a positive bead of type 1 and the right-most bead of s_n is a positive bead of type n.

The condition of the theorem ensures that neither s'_1 nor s'_n are empty. Consider the "big" necklace $s' = s'_1\bar{s}_2 s_3 \bar{s}_4 \cdots$. If n is even, the big necklace s' ends with $\bar{s}'_n$, and if n is odd, it ends with s'_n. (Given a sequence t of beads, the notation $\bar{t}$ means t where all positive beads become negative and conversely, without changing their types.) According to Proposition 2, there is a splitting of s' into $n+1$ subnecklaces $t_1, \ldots, t_{n+1}$ (completing with zero-length subnecklaces if necessary) such that we have for $i \in [n]$ in Case (i) and for $i \in \{2, \ldots, n-1\}$ in Case (ii)

$$\sum_{k \text{ odd}} \mu_i(t_k) = \sum_{k \text{ even}} \mu_i(t_k) \ , \tag{2}$$

and such that in Case (ii)

$$\sum_{k \text{ odd}} \mu_1(t_k) = \sum_{k \text{ even}} \mu_1(t_k) - 1 \ , \tag{3}$$

$$\sum_{k \text{ odd}} \mu_n(t_k) = \sum_{k \text{ even}} \mu_n(t_k) + 1 \ . \tag{4}$$

Here, we denote by $\mu_i(t)$ the amount of beads of type i in the subnecklace t. We remind the reader that the amount of beads is the number of positive beads minus the number of negative beads of this type.

We interpret now the endpoints of the subnecklaces t_k as cuts of the "big" necklace s (the one formed by the original necklaces s_j). Together with the endpoints of the s_j, we get a splitting of s into $2n$ subnecklaces $u_1, \ldots, u_{2n}$ (in this order). Some u_ℓ can be zero-length subnecklaces. We make two parts: a part A formed by the u_ℓ with an odd ℓ, and a part B formed by the u_ℓ with an even ℓ. Remark two things:

- Since the left-most bead of s_1 belongs to u_1 and the right-most bead of s_n belongs to u_{2n}, each part contains at least one bead.
- The number of subnecklaces u_ℓ is the same in each part, and thus equal to n.

We finish the proof by checking that both A and B are balanced. We denote by q_i^A and q_i^B the amount of beads of type i in A and B, respectively. Since the original collection is balanced, we have $q_i^A + q_i^B = 0$. The end of the proof consists simply in checking that we have also $q_i^A - q_i^B = 0$, which implies then immediately that $q_i^A = q_i^B = 0$, as desired.

Each bead x belongs to exactly one s_j. Moreover, apart from the two beads at the endpoints of s, any bead x (resp. $\bar{x}$) belongs also to exactly one t_k when j is odd (resp. even). It is immediate to check that $j + k - 1$ and the index ℓ of the u_ℓ to which x belongs have the same parity. Thus if $j + k - 1$ is odd, then x belongs to A, and if it is even, then x belongs to B. We have

$$q_1^A = 1 + \sum_{j,k \text{ odd}} \mu_1(s_j \cap t_k) - \sum_{j,k \text{ even}} \mu_1(\bar{s}_j \cap t_k) \text{ and } q_1^B = \delta_{\text{(i)}} + \sum_{\substack{j \text{ odd} \\ k \text{ even}}} \mu_1(s_j \cap t_k) - \sum_{\substack{j \text{ even} \\ k \text{ odd}}} \mu_1(\bar{s}_j \cap t_k) \ ,$$

and for $i \neq 1$, we have

$$q_i^A = \sum_{j,k \text{ odd}} \mu_i(s_j \cap t_k) - \sum_{j,k \text{ even}} \mu_i(\bar{s}_j \cap t_k) \quad \text{and} \quad q_i^B = \delta_{\text{(ii)}}^{i=n} + \sum_{\substack{j \text{ odd} \\ k \text{ even}}} \mu_i(s_j \cap t_k) - \sum_{\substack{j \text{ even} \\ k \text{ odd}}} \mu_i(\bar{s}_j \cap t_k) \ ,$$

where $\delta_{(\mathrm{i})} \in \{0,1\}$ and takes the value 1 only if we are in Case (i), and where $\delta_{(\mathrm{ii})}^{i=n} \in \{0,1\}$ and takes the value 1 only if we are in Case (ii) and $i = n$.

For the beads of type 1, we have

$$\begin{aligned} q_1^A - q_1^B &= 1 - \delta_{(\mathrm{i})} + \sum_{k=1}^{n+1} \sum_{j \text{ odd}} (-1)^{k+1} \mu_1(s_j \cap t_k) + \sum_{k=1}^{n+1} \sum_{j \text{ even}} (-1)^{k+1} \mu_1(\bar{s}_j \cap t_k) \\ &= 1 - \delta_{(\mathrm{i})} + \sum_{k=1}^{n+1} (-1)^{k+1} \mu_1(t_k) \\ &= 0 \ , \end{aligned}$$

where the last equality is a consequence of Equation (2) in Case (i) and of Equation (3) in Case (ii). For the beads of type $i \neq 1$, we have

$$\begin{aligned} q_i^A - q_i^B &= -\delta_{(\mathrm{ii})}^{i=n} + \sum_{k=1}^{n+1} \sum_{j \text{ odd}} (-1)^{k+1} \mu_i(s_j \cap t_k) + \sum_{k=1}^{n+1} \sum_{j \text{ even}} (-1)^{k+1} \mu_i(\bar{s}_j \cap t_k) \\ &= -\delta_{(\mathrm{ii})}^{i=n} + \sum_{k=1}^{n+1} (-1)^{k+1} \mu_i(t_k) \\ &= 0 \ , \end{aligned}$$

where the last equality is a consequence of Equation (2), except when $i = n$ and we are in Case (ii), where we use of Equation (4) instead. □

5. Alternate combinatorial proofs of Proposition 2 and Lemma 1

All statements deal with discrete objects and properties. It is thus desirable from a logical point of view that all proofs stay in the "discrete world" if possible. Similarly to the traditional proof of the necklace splitting theorem, the proof of Proposition 2 we gave in Section 4 is relying on some "continuous" notions. Pálvőlgyi [Pál09] showed how such proofs are amenable to the discrete world by using a combinatorial counterpart of the Borsuk–Ulam theorem.

We adapt here the approach proposed by Pálvőlgyi to provide a purely combinatorial proof of Proposition 2, also relying on Tucker's lemma. Moreover, we show how the more general Ky Fan lemma can actually provide a direct combinatorial proof of Lemma 1 itself.

5.1. **Combinatorial tools.** We start by introducing some notation. Let $\mathbb{B} = \{-1, 1\}$. For brevity, we may write $+$ instead of $+1$ and $-$ instead of -1. Let $\mathbb{O} = \{-1, 0, 1\}$. We define the partial order $\prec$ on $\mathbb{O}$ where $0 \prec 1$ and $0 \prec -1$, or, equivalently, $b \preccurlyeq b'$ if $b = 0$ or $b = b'$. For every $m \in \mathbb{N}$, we lift $\prec$ to $\mathbb{O}^m$ where for $x, y \in \mathbb{O}^m$ we have $x \preccurlyeq y$ if for all $i \in [m]$, $x(i) \preccurlyeq y'(i)$. Given a string x, we denote by $x(i)$ the i-th letter of x. We also denote by $-x$ the string obtained from x by replacing 1's by -1's and -1's by 1's (and the 0's are left unchanged).

Now the Ky Fan lemma can be stated as follows:

Lemma 2 (Octahedral Ky Fan lemma, [Che11, Lemma 2])**.** *Let* $\lambda \colon \mathbb{O}^m \setminus 0^m \to \{\pm 1, \dots, \pm q\}$ *such that*

(i) $\lambda(-x) = -\lambda(x)$ *for every* x.
(ii) $\lambda(x) + \lambda(y) \neq 0$ *for every* $x \preccurlyeq y$.

Then there is at least one positively alternating m-chain, i.e., there are $x_1, \dots, x_m \in \mathbb{O}^m \setminus 0^m$ and $j_1, \dots, j_m \in [q]$ such that $x_1 \preccurlyeq \cdots \preccurlyeq x_m$, $1 \leqslant j_1 < j_2 < \cdots < j_m$, and $\lambda(\{x_1, \dots, x_m\}) = \{j_1, -j_2, j_3, \dots, (-1)^{m-1} j_m\}$. In particular, $q \geqslant m$.

(The statement we provide here is actually a special case of the original Ky Fan lemma [Fan52] when the simplicial complex is the first barycentric subdivision of the octahedron.) The octahedral Tucker lemma [Tuc46] is actually the same statement without the existence of the chain, but still with the inequality $q \geqslant m$.

We explain now how strings in $\mathbb{O}^m$ relate to decompositions of strings in Σ_n^m. We consider here all decompositions of a string $w \in \Sigma_n^m$ into two tuples $(u_1, u_3, \dots)$, $(u_2, u_4, \dots)$ of strings in Σ_n^+ such that $w = u_1 u_2 \cdots$. Such a decomposition is described as a string $x \in \mathbb{B}^m$ where each maximal segment of consecutive -1's, as well as each maximal segment of consecutive $+1$'s, corresponds to one u_j. Each sign change in x corresponds thus to a cut in w and to a transition from a u_j to u_{j+1}. A string $x \in \mathbb{O}^m$, containing some 0's, can be interpreted as an underspecified decomposition, which can be completed into different decompositions by replacing the 0's with -1's or $+1$'s.

5.2. **Combinatorial proof of Proposition 2.** We prove Proposition 2 in a way that is similar to [Pál09].

For x in $\mathbb{B}^*$ we let $\mathrm{alt}(x)$ be the number of sign alternations in x. We extend the function alt to $\mathbb{O}^* \setminus 0^*$ as follows: $\mathrm{alt}(x) = \max\{\mathrm{alt}(y) \mid y \in \mathbb{B}^*,\ x \preccurlyeq y\}$. Notice that:

- $\mathrm{alt}(-x) = \mathrm{alt}(x)$ for every x.
- $\mathrm{alt}(x) \geqslant \mathrm{alt}(y)$ for every $x \preccurlyeq y$.

For x in $\mathbb{O}^* \setminus 0^*$, we define $\mathrm{sign}(x) \in \{-1, 1\}$ as the first letter of some y in $\mathbb{B}^*$ so that $x \preccurlyeq y$ and $\mathrm{alt}(x) = \mathrm{alt}(y)$. The function $\mathrm{sign}(x)$ is well defined as when x is of the form $0^k \kappa x'$, with $\kappa \in \mathbb{B}$, the only way to replace the first k zeroes so as to maximize alt consists in changing the sign at each position.

From now on we fix the necklace w and assume it is of length m (i.e., w is in Σ_n^m).

We let $E_{\kappa,i}(x)$ for x in $\mathbb{O}^m \setminus 0^m$ and $\kappa \in \{-1, 1\}$ be the amount of beads of type i in w that are aligned with the symbol κ in x. We say that x is *κi-unbalanced* if $E_{\kappa,i}(y) > E_{-\kappa,i}(y)$ for every y such that $x \preccurlyeq y$. We define $\mathrm{unb}(x)$ to be κi where i is the smallest i so that x is κi-unbalanced. When no such i exists, we let $\mathrm{unb}(x) = 0$. We have

- $\mathrm{unb}(-x) = -\mathrm{unb}(x)$ for every x.
- $|\mathrm{unb}(x)| \geqslant |\mathrm{unb}(y)| > 0$ for every $x \preccurlyeq y$ if $\mathrm{unb}(x) \neq 0$.

Notice that $|\mathrm{unb}(x)| = |\mathrm{unb}(y)|$ implies $\mathrm{unb}(x) = \mathrm{unb}(y)$ for every $x \preccurlyeq y$. Finally we define λ as the function from $\mathbb{O}^m$ to $\{-m+1, \dots, 0, \dots, m-1\}$:

$$\lambda(x) = \begin{cases} \mathrm{sign}(x)\,\mathrm{alt}(x) & \text{when } \mathrm{alt}(x) > n \\ \mathrm{unb}(x) & \text{when } \mathrm{alt}(x) \leqslant n \end{cases}.$$

Because of the properties of alt, sign and unb, had it no zero, the function λ would satisfy the properties (i) and (ii) of the octahedral Tucker lemma. However, then the conclusion is not satisfied, thus λ must have a zero. Take x so that $\lambda(x) = 0$. We must have that $\mathrm{alt}(x) \leqslant n$. As $\mathrm{unb}(x) = 0$, it is possible, for each type i, to replace 0's in x with 1's or -1's so that we eventually obtain y verifying $E_{1,i}(y) = E_{-1,i}(y)$ for every i, and such that the remaining 0's in y are aligned with at most one unassigned bead of type i. We can then choose to replace these 0's with either 1 or -1 depending on how we wish to treat the extra bead of a

given type. As $\mathrm{alt}(x)$ is smaller than n, then no matter how we have conducted the previous changes, we have obtained a way to split the necklace with at most n cuts so that each part contains the same amount of each type, with the extra beads (when the amount is odd) being distributed in any possible way.

Remark. The definition of unb is where the proof departs from the one of [Pál09]. Indeed Pálvőlgyi's proof only considers positive beads and it is then enough to consider that x is unbalanced when one of the thief receives more than half of the beads of some type. In that case the split remains unbalanced for every y so that $x \preccurlyeq y$. This property is no longer true with negative beads. Our definition of unb is taking this into account by imposing that being unbalanced is closed under $\preccurlyeq$.

5.3. **Combinatorial proof of Lemma 1.** Lastly, we present a direct combinatorial proof of Lemma 1 that avoids Theorem 2. The method is again inspired by Pálvőlgyi [Pál09], but instead of constructing a function λ that contradicts the Tucker lemma, if it has no zero, we construct a pair of functions λ_+ and λ_-, for which the chains guaranteed by the Ky Fan lemma cannot exist for irreducible tuples.

In contrast to Section 5.2, we do not consider $x \in \mathbb{B}^m$ to represent the decomposition of a single string but instead x shall represent the decomposition of an irreducible tuple $(s_1, \dots, s_n)$ in $(\Sigma_n^+)^n$ where the word $s_1 \cdots s_n$ is in O_n and each s_j is of length at least 2. To account for the arity of the tuple, we need to generalize the notion of alt. Denote the word $s_1 \cdots s_n$ by s and the length of s by m. Factorize x such that $x = \kappa_1^{\ell_1}\kappa_2^{\ell_2}\cdots\kappa_k^{\ell_k}$ with $\kappa_1, \dots, \kappa_k \in \mathbb{B}$, $\kappa_p = -\kappa_{p+1}$ and $\ell_p \in \mathbb{N}$ be such that for each $j \in [n]$, there is $q_j \in \{0, \dots, k\}$ with $\sum_{p\in[j]} |s_p| = \sum_{p\in[q_j]} \ell_p$, and k is minimal. Now, if $k = 2n$, then x describes a decomposition of $(s_1, \dots, s_n)$ in two tuples each of size n. Precisely, these tuples are $(u_1, u_3, \dots, u_{2n-1})$ and $(u_2, u_4, \dots, u_{2n})$ where the length of u_j is ℓ_j for each $j \in [2n]$ and $s_j = u_{q_{i-1}+1}u_{q_{i-1}+2}\cdots u_{q_j}$ for each $j \in [n]$. If $k < 2n$, we may add κ_q's with $\ell_q = 0$ to increase k to $2n$ and proceed similarly. We call the value $k-1$ the number of *sign alternations of* x and write $\mathrm{alt}(x) = k-1$. Hence, in the following we will search for x with $\mathrm{alt}(x) \leqslant 2n-1$. Crucially, we assume mandatory sign alternations between the *neighboring endpoints* of s_j and s_{j+1}: if the last position of s_j and the first position of s_{j+1} are signed equally in x, then some κ_p with $\ell_p = 0$ occurs in our factorization of x. In this case 2 sign alternations are accounted for the transition between neighboring endpoints. In order to maximize alt, we may choose x to be strictly

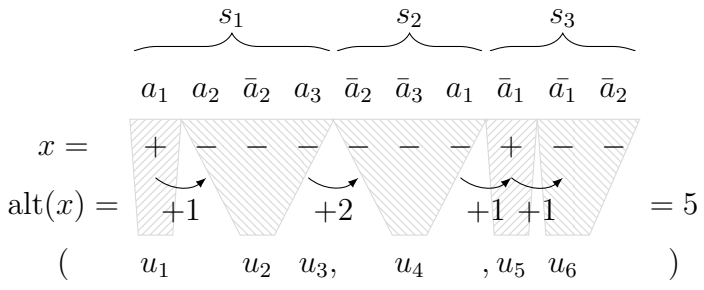

FIGURE 1. A word $s_1s_2s_3$, a decomposition x with the corresponding alignment of s_j's to u_p's (in particular, $u_3 = \varepsilon$), and the calculation of $\mathrm{alt}(x)$.

alternating except for the $n-1$ transitions between neighboring endpoints. In consequence $\mathrm{alt}(x) \leqslant m-(n-1)+2(n-1)-1 = m+n-2$.

Figure 1 gives an example: Note that $\mathrm{alt}(x) < 2 \cdot 3$ and that the subwords in $+$-labeled and $-$-labeled positions are both in O_3. Thus, x describes the decomposition of $(a_1a_2\bar{a}_2a_3, \bar{a}_2\bar{a}_3a_1, \bar{a}_1\bar{a}_1\bar{a}_2)$ to two strictly smaller 3-tuples $(a_1, \varepsilon, \bar{a}_1)$ and $(a_2\bar{a}_2a_3, \bar{a}_2\bar{a}_3a_1, \bar{a}_1\bar{a}_2)$, for each of which the word obtained by concatenating the tuple's components is in O_3.

Again, we consider words in $\mathbb{O}^m \setminus 0^m$ which may have *unsigned* positions, i.e., positions labeled with 0. In this case the decomposition is only partially determined. We define the function $h\colon \mathbb{O}^m \setminus 0^m \to [m+n-2]$ by $h(x) = \max_{y \in \mathbb{B}^m\colon x \preccurlyeq y} \mathrm{alt}(y)$. Observe that $\mathrm{alt}(x) = \mathrm{alt}(-x)$, hence, for each $x \in \mathbb{O}^m \setminus 0^m$: $h(x) = h(-x)$. Also, for each $x, y \in \mathbb{O}^m \setminus 0^m$ with $x \preccurlyeq y$, we have $h(x) \geqslant h(y)$.

Denote by $H(x)$ the set $\arg\max_{y \in \mathbb{B}^m\colon x \preccurlyeq y} \mathrm{alt}(y)$. All elements of $H(x)$ may be obtained with the following algorithm:

(i) Choose a signed position p in x with an unsigned neighbor $p' \in \{p-1, p+1\}$.
(ii) If p and p' are neighboring endpoints (in this case we call p and p' *internal endpoints*), set $x'(p') = x(p)$; otherwise set $x'(p') = -x(p)$. For each $q \neq p'$, set $x'(q) = x(q)$.
(iii) If $x' \in \mathbb{B}^m$ we are done, otherwise, recursively apply the algorithm to x'.

This algorithm may yield words y, y' where $y(p) \neq y'(p)$ if p is an unsigned position between two signed positions of x. However, for all position p' smaller than the smallest signed position of x, the signs $y(p')$ and $y'(p')$ are equal for all words y, y' in $H(x)$. Hence, we may define a function $\mathrm{sign}\colon \mathbb{O}^m \setminus 0^m \to \mathbb{B}$ to assign to x the first symbol of some $y \in H(x)$. Moreover, for each $x \in \mathbb{O}^m \setminus 0^m$: $\mathrm{sign}(x) = -\mathrm{sign}(-x)$.

To prove Lemma 1, we need to show that there is $x \in \mathbb{B}^m$ with $\mathrm{alt}(x) < 2n$ and that both words $u_1u_3\cdots$ and $u_2u_4\cdots$ described by x are in O_n and non-empty. To assure non-emptiness, we just need that both signs occur in x. To ensure that the words are in O_n, we rephrase the notions $E_{\kappa,i}$ and unb from Section 5.2 in terms of tuples of words instead of necklaces. Now $E_{\kappa,i}(x)$ is formulated as the difference of the amount of a_i's and $\bar{a}_i$'s aligned with κ in x. Formally:

$$E_{\kappa,i}(x) = |\{p \mid s(p) = a_i \wedge x(p) = \kappa\}| - |\{p \mid s(p) = \bar{a}_i \wedge x(p) = \kappa\}| \ .$$

Observe that $E_{+,i}(x) + E_{-,i}(x) = 0$ for $s \in O_n$ and $x \in \mathbb{B}^m$. We define two functions λ_+ and λ_-, where for each $b \in \mathbb{B}$, $\lambda_b\colon \mathbb{O}^m \setminus 0^m \to \{\pm 1, \ldots, \pm m\} \cup \{0\}$ is such that

$$\lambda_b(x) = \begin{cases} \mathrm{sign}(x) \cdot (h(x) - n + 2) & \text{if } h(x) \geqslant 2n \ . & \text{(C1)} \\ b \cdot (-1) \cdot b' \cdot (n+1) & \text{if } h(x) < 2n \wedge \exists b' \in \mathbb{B}\colon |x|_{b'} = 0 \ . & \text{(C2)} \\ \mathrm{unb}(x) & \text{if } h(x) < 2n \wedge |x|_+ > 0 \wedge |x|_- > 0 \ . & \text{(C3)} \end{cases}$$

If there are $x \in \mathbb{O}^m \setminus 0^m$ and $b \in \mathbb{B}$ such that $\lambda_b(x) = 0$, then Lemma 1 holds: Note that case C3 of λ_b applies. Thus, for each $i \in [n]$ there is $y \in \mathbb{B}^m$ such that $E_{+,i}(y) = E_{-,i}(y) = 0$. The positions relevant to balance a_i and a_j where $i \neq j$ are distinct. Hence, we choose $\hat{y} \in \mathbb{B}^m$ with $x \preccurlyeq \hat{y}$ such that all symbols are balanced. Since $\hat{y} \succcurlyeq x$ we have that $\mathrm{alt}(\hat{y}) \leqslant h(x) < 2n$, $|\hat{y}|_+ > 0$, and $|\hat{y}|_- > 0$. Thus, $\hat{y}$ encodes a decomposition with the desired properties.

Otherwise, if λ_+ and λ_- have no zero, we want to apply the Ky Fan lemma. Clearly, λ_b satisfies property (i) of the octahedral Ky Fan lemma if cases C1 and C3 apply. For case C2, note that $|x|_+ = 0 = |-x|_-$ implies

$$-\lambda_b(x) = -(b \cdot (-1) \cdot (+1) \cdot (n+1)) = (b \cdot (-1) \cdot (-1) \cdot (n+1)) = \lambda_b(-x) \ .$$

It remains to show property (ii), i.e., that the sum $\lambda_b(x) + \lambda_b(y)$ is not zero if $x \preccurlyeq y$ and $\lambda_b(x) \neq 0 \neq \lambda_b(y)$. Assume $\lambda_b(x) + \lambda_b(y) = 0$. By the definition of λ_b, there are three cases:

- C1: $h(x) = h(y) \geqslant 2n$ and $\operatorname{sign}(x) = -\operatorname{sign}(y)$. But then $H(x) \supseteq H(y)$ and, thus, $\operatorname{sign}(x) = \operatorname{sign}(y)$, a contradiction.
- C2: $h(x), h(y) < 2n$ and either $|x|_+ = 0 = |y|_-$ or $|x|_- = 0 = |y|_+$. W.l.o.g. assume that $|x|_- = 0 = |y|_+$. Then there is $k \in [m]$ such that $x(k) = +$. But $x \preccurlyeq y$ implies $y(k) = +$, a contradiction to $|y|_+ = 0$.
- C3: $h(x), h(y) < 2n$, $|x|_+, |y|_+ > 0$, $|x|_-, |y|_- > 0$ and $\operatorname{unb}(x) = -\operatorname{unb}(y)$. This contradicts properties of unb shown in Section 5.2: for every $x \preccurlyeq y$ with $\operatorname{unb}(x) \neq 0$ we have $|\operatorname{unb}(x)| \geqslant |\operatorname{unb}(y)|$, where $|\operatorname{unb}(x)| = |\operatorname{unb}(y)|$ implies $\operatorname{unb}(x) = \operatorname{unb}(y)$.

Proposition 3. *If λ_+ and λ_- have no zero, then every pair of compatible symbols $a_i, \bar{a}_i \in \Sigma_n$ can be reduced from $(s_1, \ldots, s_n)$.*

Lemma 1 follows directly, as whenever $(s_1, \ldots, s_n)$ is irreducible, λ_+ and λ_- have a zero.

To show Proposition 3 assume that for either $b \in \mathbb{B}$ there is no $x \in \mathbb{O}^m \setminus 0^m$ such that $\lambda_b(x) = 0$. By Lemma 2, for each $b \in \mathbb{B}$, there is a positively alternating m-chain, i.e., there are $x_1^b \preccurlyeq x_2^b \preccurlyeq \cdots \preccurlyeq x_m^b$ and $1 \leqslant j_1^b < j_2^b < \cdots < j_m^b \leqslant m$ such that $\lambda_b(\{x_1^b, \ldots, x_m^b\}) = \{(-1)^{k-1} j_k^b \mid k \in [m]\}$. We derive the following properties of the chains and $(s_1, \ldots, s_n)$:

(a) The values $j_1^b, \ldots, j_m^b$ are pairwise distinct and in $[m]$. Thus, $j_k^b = k$ for each $k \in [m]$.

(b) By the definition of $\prec$, every strictly increasing chain in $\mathbb{O}^m \setminus 0^m$ has at most length m. Specifically, $|x_1^b|_+ + |x_1^b|_- = 1$, $|x_2^b|_+ + |x_2^b|_- = 2$, …, and $|x_m^b|_+ + |x_m^b|_- = m$. Also, for each $k \in [m-1]$, x_k^b and x_{k+1}^b differ in exactly one position p, i.e., $x_k^b(p) = 0$ and $x_{k+1}^b(p) \in \mathbb{B}$ and for each $p' \neq p$: $x_k^b(p') = x_{k+1}^b(p')$.

(c) By (a), all cases (C1, C2, C3) of λ_b apply. Because h is antitone, C1 applies to $x_1^b, \ldots, x_{m-n-1}^b$ where $h(x_k^b) > h(x_{k+1}^b)$ for each $k \in [m-n-1]$. On the other hand, for $k \in \{0, 1, \ldots, n\}$, we have that $h(x_{m-n+k}^b) < 2n$.

Case C2 applies to exactly one of the words $x_{m-n}^b, \ldots, x_m^b$ while to the remaining words case C3 applies. For C2 to apply, only one sign $\top_b \in \mathbb{B}$ may occur in the word, while C3 requires that both signs occur. Hence C2 applies to x_{m-n}^b as the words, to which C3 apply, need to be larger with respect to $\preccurlyeq$. Moreover, as x_{m-n}^b is larger than $x_1^b, \ldots, x_{m-n-1}^b$, the latter are in $\{\top_b, 0\}^m$, too. $\top_b$ is determined by the sign of $\lambda_b(x_{m-n}^b) = (-1)^n \cdot (n+1)$, i.e., it depends on b and the parity of n. Precisely: $\top_b = b \cdot (-1)^n$.

(d) C3 applies to x_{m-n+k}^b for each $k \in [n]$ and yields values $\{1, -2, 3, \ldots, (-1)^{n-1} n\}$. As $\operatorname{unb}(x_{m-n+k}^b) \neq 0$, we have that $|\operatorname{unb}(x_{m-n+k}^b)| > |\operatorname{unb}(x_{m-n+k+1}^b)|$ for all $k \in [n-1]$. Thus, $\lambda(x_{m-n+k}^b) = (-1)^{n-k} \cdot (n-k+1)$. Denote $n-k+1$ by i. Observe that $\operatorname{unb}(x_{m-n}^b) = 0$ because the word ${\top_b}^m \succcurlyeq x_{m-n}^b$ trivially balances every symbol. Hence, to ultimately unbalance a_i, a position p_i^b is newly signed in x_{m-n+k}^b with $-\top_b$ where $s(p_i^b)$ is either a_i or $\bar{a}_i$. The exact choice of p_i^b depends on $\top_b$ and the parity of i: If $(-1)^{i-1} = -\top_b$, then p_i^b is such that $s(p_i^b) = a_i$, otherwise, p_i^b is such that $s(p_i^b) = \bar{a}_i$.

Let $U = \{p \in [m] \mid \exists b \in \mathbb{B}\colon x_{m-n}^b(p) = 0\} = \{p_i^b \mid b \in \mathbb{B}, i \in [n]\}$. From

$$\top_+ = (+1) \cdot (-1)^n = (-1) \cdot (-1) \cdot (-1)^n = (-1) \cdot \top_-$$

and (d) it follows that $p_i^+ \neq p_i^-$ and $a_i, \bar{a}_i \in \{s(p) \mid p \in U\}$ for each $i \in [n]$. Hence, $|U| = 2n$ and $\Sigma_n \subseteq \{s(p) \mid p \in U\}$. To complete our proof, we further characterize the set U.

Property 1. *Let p and $p+1$ be neighboring endpoints. At most one of p and $p+1$ is signed in x^b_{m-n}.*

Proof. Assume the contrary. Let $k \in [m-n]$ be minimal such that $x^b_k(p) = \top_b = x^b_k(p+1)$, i.e., either $x^b_{k-1}(p) = 0$ or $x^b_{k-1}(p+1) = 0$. W.l.o.g. assume that $x^b_{k-1}(p+1) = 0$. Hence, using our algorithm to construct an element of $H(x^b_{k-1})$, we may set $p+1$ to $\top_b$, i.e., we obtain x^b_k. But then $h(x^b_k) = h(x^b_{k-1})$, contradicting (c). □

Let p and $p+1$ be neighboring endpoints. We denote $C(p) = C(p+1) = \{p, p+1\}$. For each position $p' \in [m]$ that is not an internal endpoint, we denote $C(p') = \{p'\}$.

Property 2. *Let $1 < p < m$ be an unsigned position in x^b_{m-n} that is not an internal endpoint. Then $C(p-1)$ or $C(p+1)$ contain only unsigned positions in x^b_{m-n}.*

Proof. Assume the contrary. Let $k \in [m-n]$ be minimal such that both $C(p-1)$ and $C(p+1)$ contain a signed position in x^b_k. W.l.o.g. assume that $C(p-1)$ contains a signed position already in x^b_{k-1}. Using the algorithm to construct an element of $H(x^b_{k-1})$, we may first set the other position in $C(p-1)$ to $\top_b$ (if it exists), then set p to $-\top_b$, then set each position of $C(p+1)$ to $\top_b$ and call the intermediate result y. We may also obtain y from x^b_k by first setting each unsigned position of $C(p+1)$ to $\top_b$, then p to $-\top_b$, and then the remaining position of $C(p-1)$ to $\top_b$, if it exists. Thus, $h(x^b_k) = h(x^b_{k-1})$, contradicting (c). □

There are $n-1$ pairs of neighboring endpoints, i.e., in total $2n-2$ internal endpoints. x^+_{m-n} and x^-_{m-n} each have n unsigned positions, of which, by Property 1, at least $n-1$ are internal endpoints. Property 2 implies that the only unsigned position that is not an internal endpoint, if one exists, is 1 or m. Thus, U consists solely of endpoints of the s_j's. However, all symbols of Σ_n are distributed at the positions in U. Hence, $(s_1, \ldots, s_n)$ is such that every pair of compatible letters of Σ_n can be reduced. This concludes the proof of Proposition 3.

References

[AMS06] Noga Alon, Dana Moshkovitz, and Shmuel Safra. Algorithmic construction of sets for k-restrictions. *ACM Transactions on Algorithms (TALG)*, 2(2):153–177, 2006.

[AW86] Noga Alon and Douglas B. West. The Borsuk–Ulam theorem and bisection of necklaces. *Proceedings of the American Mathematical Society*, 98(4):623–628, 1986.

[Bac81] Emmon Bach. Discontinuous constituents in generalized categorial grammars. In Victoria Burke and James Pustejovsky, editors, *Proceedings of the 11th Annual Meeting of the Northeastern Linguistics Society*, pages 1–12, 1981.

[Bac88] Emmon Bach. Categorial grammars as theories of language. In Richard T. Oehrle, Emmon Bach, and Deirdre Wheeler, editors, *Categorial Grammars and Natural Language Structures*, pages 17–34. D. Reidel, 1988.

[Che11] Peng-An Chen. A new coloring theorem of Kneser graphs. *Journal of Combinatorial Theory, Series A*, 118(3):1062–1071, 2011.

[Fan52] Ky Fan. A generalization of Tucker's combinatorial lemma with topological applications. *Annals of Mathematics*, 56(3):431–437, 1952.

[Gil05] Robert Gilman. Formal languages and their application to combinatorial group theory. In *Groups, Languages, Algorithms*, number 378 in Contemporary Mathematics, pages 1–36. Amer. Math. Soc., 2005.

[Ho18] Meng-Che Ho. The word problem of $\mathbb{Z}_n$ is a multiple context-free language. *Groups Complexity Cryptology*, 10(1):9–15, 01 May. 2018.

[HR65] Charles R. Hobby and John R. Rice. A moment problem in l_1 approximation. *Proceedings of the American Mathematical Society*, 16(4):665–670, 1965.

[Jos85] Aravind K. Joshi. Tree-adjoining grammars: How much context sensitivity is required to provide reasonable structural descriptions? In David Dowty, Lauri Karttunen, and Arnold M. Zwicky, editors, *Natural Language Parsing*, pages 206–250. Cambridge University Press, 1985.

[JSW91] Aravind K. Joshi, Vijay K. Shanker, and David J. Weir. The converence of mildly context-sensitive grammar formalisms. In Peter Sells, Stuart M. Shieber, and Thomas Wasow, editors, *Foundational Issues in Natural Language Processing*, pages 31–81. The MIT Press, 1991.

[KS12] Makoto Kanazawa and Sylvain Salvati. MIX is not a tree-adjoining language. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 666–674, 2012.

[MS83] David E. Muller and Paul E. Schupp. Groups, the theory of ends, and context-free languages. *Journal of Computer and System Sciences*, 26(3):295–310, 1983.

[Ned16] Mark-Jan Nederhof. A short proof that O_2 is an MCFL. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016.

[Ned17] Mark-Jan Nederhof. Free word order and MCFLs. In M. Wieling, M. Kroon, G. van Noord, and G. Bouma, editors, *From Semantics to Dialectometry: Festschrift for John Nerbonne*, chapter 28, pages 273–282. College Publications, 2017.

[Pál09] Dömötör Pálvölgyi. Combinatorial necklace splitting. *The Electronic Journal of Combinatorics*, pages R79–R79, 2009.

[Pin76] Allan Pinkus. A simple proof of the Hobby–Rice theorem. *Proceedings of the American Mathematical Society*, 60(1):82–84, 1976.

[Pul83] Geoffrey K. Pullum. Context-freeness and the computer processing of human languages. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, pages 1–6, 1983.

[Sal15] Sylvain Salvati. MIX is a 2-MCFL and the word problem in is captured by the IO and the OI hierarchies. *Journal of Computer and System Sciences*, 81(7):1252–1277, 2015.

[SK08] Hiroyuki Seki and Yuki Kato. On the generative power of multiple context-free grammars and macro grammars. *IEICE Transactions*, 91-D(2):209–221, 2008.

[SMFK91] Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. On multiple context free grammars. *Theoretical Computer Science*, 88(2):191–229, 1991.

[Tuc46] Albert W. Tucker. Some topological properties of disk and sphere. In *Proc. First Canadian Math. Congress 1945*, pages 285–309, Montreal, 1946. University of Toronto Press.

[Wei88] David J. Weir. *Characterizing mildly context-sensitive grammar formalisms*. PhD thesis, University of Pennsylvania, Philadephia, PA, 1988.