Picture Hanging Puzzle Solution

ETHZ Math Olympiad Club

15 September 2025

Problem (Picture Hanging Puzzle, A. Spivak, 1997)

We have $n \geq 1$ nails fixed to a wall and a sufficiently long rope wrapped around these nails. We consider here what we call *oriented wrapping*, which means a circuit (with a direction) of the rope that may loop around the nails (either clockwise or counterclockwise). To help you imagine the situation, consider the two endpoints of the rope carrying a framed picture. We strongly encourage you to try to recreate the scenario in the real world with a sufficiently long rope and some nails or pins fixed to a surface.

Consider all oriented wrappings around the n nails satisfying the following two conditions:

1. The circuit is non-trivial; that is, the rope remains securely wrapped (i.e., it does not fall off when all nails are present), or, equivalently, the framed picture does not fall.
2. When **any single nail** is removed (regardless of which one), the entire rope falls from the wall. In practice, some friction might prevent it from falling, but we consider it as "falling" if it is no longer securely wrapped.

Here are examples where we have labelled the nails c_0, c_1, and c_2:

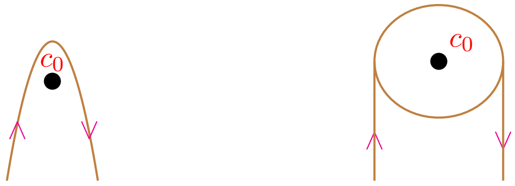

Figure 1: *Oriented wrappings for $n = 1$* ***satisfying*** *the properties. The first configuration has a single loop around c_0 (clockwise), while the second has at least two clockwise loops around c_0.*

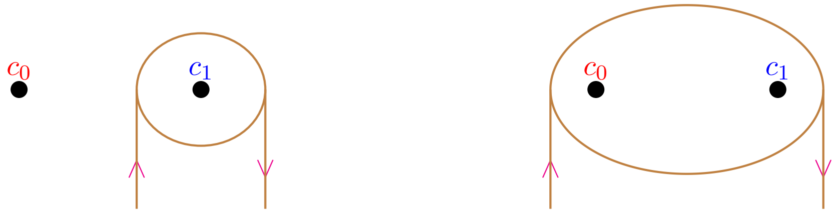

Figure 2: *Non-trivial oriented wrappings for $n = 2$* ***not satisfying*** *the property 2. The first configuration has no loop around c_0 but at least two clockwise loops around c_1. The second configuration has at least two clockwise loops around both c_0 and c_1.*

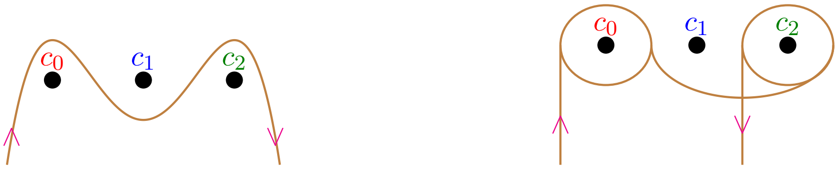

Figure 3: *Nontrivial oriented wrappings for $n = 3$* ***not satisfying*** *the property* 2*. The first configuration has a single clockwise loop around c_0, no loop around c_1, and a single clockwise loop around c_2. The second configuration has at least two clockwise loops around c_0, no loop around c_1, and at least two counter-clockwise loops around c_2.*

For any oriented wrapping, we define its *length* to be the number of loops in its circuit. Therefore, the lengths of the oriented wrappings in the above example are, respectively (for certain $a, b, c, d, e \geq 2$):

$$1,\ a,\ b,\ 2c,\ 2,\ d+e.$$

For each $n \geq 1$, construct an oriented wrapping with the above two properties whose length is bounded by a polynomial in n, more precisely, of length at most $4n^2$.

Solution:

This problem is commonly referred to as the *picture hanging puzzle* and was originally posed by A. Spivak in 1997. A detailed account of the history of this problem, which is generalised in the paper by Demaine et al. [1], can be found there. In the paper, our problem is referred to as the 1*-out-of-n problem*. In this solutions, we first explain how the problem is commonly transposed into the natural setting of words in the free group, and then we provide a solutions of length at most $4n^2$ to the 1-out-of-n problem.

The magic behind this problem lies in formalization. Let $n \geq 1$. To formalize the mathematical model of oriented wrappings, we consider a set of n distinct symbols $\{c_i \,|\, i \in n\}$, interpreted as n nails. For simplicity, we imagine nails pinned from left to right, with c_0 on the far left and c_{n-1} on the far right. Physical intuition tells us that what **only** matters in the oriented wrapping are the orientations, number, and adjacency relationships of the loops. (Visualisation is simpler in a physical simulation!) Note that the orientation of the loops determines the order in which the wrapping was executed[1] (performed from left to right in the examples of the problem statement). Thus, an oriented wrapping corresponds to a wrapping procedure. Conversely, a wrapping procedure gives an oriented wrapping, where the direction of the orientation follows the arrow of time in the wrapping procedure.

Fix an orientation convention: clockwise as 1, counterclockwise as -1. Given an oriented wrapping, as explained, we can view it as a wrapping procedure. Passing **above** a nail from left to right counts as a clockwise loop[2]; passing from right to left counts as counterclockwise. Thus, each action is determined by the nail on which it operates and the orientation of the loop performed; therefore, we can encode it as $c_i^\theta \overset{\text{not}}{=} (c_i, \theta)$, where $i < n$ and $\theta \in \{1, -1\}$. The adjacency of the actions (which are obviously non-commutative) is represented by writing them from left to right in the same order as the execution of the wrapping procedure. Hence, we have a simple recursive procedure that, given an oriented wrapping, outputs a word over

[1] Performing it in reverse order would yield the same physical configuration. However, the wrappings here are considered oriented. If we drop the orientation condition, then the answer to the problem is taken modulo the possibility of reversing the arrow of the oriented wrapping.

[2] A "loop" need not form a closed circle, as shown in the examples in the problem statement.

the alphabet

$$C_n := \left\{ c_i^1 \,\middle|\, i < n \right\} \cup \left\{ c_i^{-1} \,\middle|\, i < n \right\},$$

by recursively recording loop orientations and positions.

More formally, if one has partially constructed a word $w \in C_n^{<\omega}$ (including the empty word $\varepsilon = \varnothing$) from a partial wrapping procedure, then we update the word as follows. If no action is taken, w remains unchanged. Otherwise, performing a loop at c_i with orientation θ updates the word to $w \frown \left\langle c_i^\theta \right\rangle$. This procedure deterministically defines a word for any oriented wrapping. The words of the oriented wrappings in the examples of the problem statement are (for some $a, b, c, d, e \geq 2$):

$$\left\langle {c_0}^1 \right\rangle, \left\langle {c_0}^1 \right\rangle^a, \left\langle {c_1}^1 \right\rangle^b, \left\langle {c_0}^1, {c_1}^1 \right\rangle^c, \left\langle {c_0}^1, {c_2}^1 \right\rangle, \left\langle {c_0}^1 \right\rangle^d \frown \left\langle {c_2}^{-1} \right\rangle^e \text{ respectively.}$$

(The powers indicate the number of times the word is concatenated with itself.) Conversely, every word over $C_n^{<\omega}$ corresponds deterministically to an oriented wrapping given by executing the symbols of the word from left to right recursively.

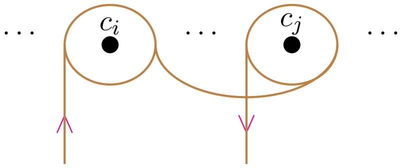

Figure 4: *If $i < j < n$, the word $\langle c_i^1 \rangle^3 \frown \left\langle c_j^{-1} \right\rangle^4$ represents: 3 clockwise loops at c_i, moving **below** intermediate nails $c_{i+1}, \ldots, c_{j-1}$ (to avoid creating unintended loops) and 4 counterclockwise loops at c_j.*

Clearly, this word-to-oriented wrapping procedure is a function for which the previous oriented wrapping-to-word procedure is a right inverse. Hence, the word-to-oriented wrapping procedure is a surjection from the set of words over the alphabet C_n, namely $C_n^{<\omega}$, to the set of oriented wrappings. This correspondence, although being a surjective functional relation, is not injective, as one may notice: there are multiple words that give the same oriented wrapping. Fix any word $w \in C_n^{<\omega}$, perform its corresponding oriented wrapping as above, then make a loop at c_i with orientation θ and then perform another loop at c_i with orientation $-\theta$; this leaves the physical state of the oriented wrapping unchanged while resulting from two different words:

$$w \frown \left\langle c_i^\theta \right\rangle \frown \left\langle c_i^{-\theta} \right\rangle \neq w.$$

This is one instance that obstructs our attempt to establish a bijective correspondence.

To be able to handle this problem comfortably, it would be good to have a bijection. Recall the general fact: if $f : A \twoheadrightarrow B$ is a surjective function, then the relation $\sim \subset A \times A$ defined by $a \sim a'$ if and only if $f(a) = f(a')$ is an equivalence relation. It induces a bijective function $\tilde{f}$ such that $\tilde{f} \circ q = f$, where $q : A \to A/_\sim$ is the quotient map.

The above discrepancies (the obstructions to injectivity) can then be resolved by identifying precisely the words that define the same oriented wrapping (using the above word-to-oriented wrapping procedure).

As observed, two words $w, w' \in C_n^{<\omega}$ yield the same oriented wrapping if, after recursively removing syntactically from w and w' every occurrence of cancelling adjacent pairs

$$\left\langle c_i^{\theta}, c_i^{-\theta} \right\rangle$$

(until no more can be removed) to obtain "clean" words v and v' from w and w' respectively, the two clean versions are equal $v = v'$.

Induction on physical intuition or algebraic topology suggests that the converse is also true: two words w, w' yielding the same oriented wrapping **must** differ by intricate combinations of cancelling adjacent pairs (that is, their clean versions, as described above, must be equal). Hence, the words that must be identified are **precisely** those that differ in this way.

To distinguish them from generic words, we will call these equivalence classes *wrapping words*. Those familiar with the subject will recognise that *wrapping words* over the alphabet C_n are exactly the elements of the free group $F\left(\{c_i \,|\, i < n\}\right)$. For the complete construction, see the appendix [A.1]. In total, we obtain a bijection: each equivalence class of a word w over C_n (i.e., each wrapping word) corresponds to a unique oriented wrapping of the rope around the nails via the word-to-oriented wrapping procedure (applied to any representative of the class). Conversely, each oriented wrapping yields a unique wrapping word by taking the equivalence class of the word obtained through the oriented wrapping-to-word procedure described above. Finally, these two procedures are inverses of one another.

With this bijective correspondence, we can reformulate the problem. We first reformulate the two required properties and the notion of length of an oriented wrapping:

- An oriented wrapping of the rope that falls, i.e., is wrapped in a trivial way, corresponds to the wrapping word $[\varepsilon]_\sim$ (the empty sequence).

- Physically removing a subset $S \subset \{c_i \,|\, i < n\}$ of the nails from an oriented wrapping corresponding to an element $W \in F(\{c_i \,|\, i < n\})$ clearly yields another oriented wrapping around the remaining nails $\{c_i \,|\, i < n\} \setminus S$. This again corresponds to an equivalence class of a word written in the same alphabet without $\left\{ s^\theta \,\middle|\, s \in S,\, \theta \in \{-1, 1\} \right\}$, that is, an element of

$$F(\{c_i \,|\, i < n\} \setminus S)\,.$$

Since

$$F(\{c_i \,|\, i < n\} \setminus S) \hookrightarrow F(\{c_j \,|\, j < n\})\,,$$

(by an easy application of the free property of $F(\{c_i \,|\, i < n\} \setminus S)$), we may view this wrapping word in $F(\{c_j \,|\, j < n\})$.

The unique function $\tilde{\pi}_S$, defined so that the following diagram commutes,

$$\begin{tikzcd}
\{\text{oriented wrappings on } \{c_i\}_{i<n}\} \arrow[r, "\text{removing } S"] \arrow[d, equal] & \{\text{oriented wrappings on } \{c_i\}_{i<n} \setminus S\} \arrow[d, hook] \\
F(\{c_j \,|\, j < n\}) \arrow[r, "\tilde{\pi}_S"'] & F(\{c_j \,|\, j < n\})
\end{tikzcd}$$

is the one induced by the free property of $F(\{c_j \,|\, j < n\})$ from the function

$$\pi_S : \{c_j \,|\, j < n\} \longrightarrow F(\{c_j \,|\, j < n\})$$

defined by (for $i < n$):

$$\pi_S(c_i) = \begin{cases} [\varepsilon]_\sim, & \text{if } c_i \in S, \\ [\langle c_i^1 \rangle]_\sim, & \text{otherwise.} \end{cases}$$

The property that the oriented wrapping of the rope falls when any single nail is removed is then equivalent to the corresponding wrapping word W lying in the kernel of each $\tilde{\pi}_{\{c_i\}}$ for $i < n$:

$$W \in \bigcap_{i<n} \ker\left(\tilde{\pi}_{\{c_i\}}\right).$$

Summarising, we obtain that the set of wrapping words corresponding to all oriented wrappings satisfying the two required properties is

$$\mathcal{S}_n := \left(\bigcap_{i<n} \ker\left(\tilde{\pi}_{\{c_i\}}\right)\right) \setminus \{[\varepsilon]_\sim\}.$$

- The *length*, length : $F(\{c_i \mid i < n\}) \to \mathbb{N}$, of a wrapping word $W \in F(\{c_i \mid i < n\})$ is defined to be the minimal domain of its representatives:

$$\operatorname{length}(W) := \min\left\{\operatorname{dom}(w) \mid w \in C_n^{<\omega},\ W = [w]_\sim\right\}.$$

We see easily that the length of an oriented wrapping is the domain of the word obtained by the oriented wrapping-to-word procedure. This word is easily seen to be clean, or equivalently, to be the word with the smallest domain among those equivalent to it. Hence, the length of an oriented wrapping **is** the length of its corresponding wrapping word.

For $n = 1$, we have clearly $\ker(\tilde{\pi}_0) = F(\{c_0\})$, hence $\mathcal{S}_1 = F(\{c_0\}) \setminus \{[\varepsilon]_\sim\}$. Indeed, it is quite intuitive (in view of the example given in the problem statement for $n = 1$) that any oriented wrapping, as long as it is not the trivial one, falls only when we remove the sole nail c_0. To answer the problem in this case, take $[\langle c_0^1 \rangle]_\sim$ (which has length $1 < 4 \cdot 1^2$).

In general, for $n \geq 2$, one might first wonder whether the intersection is trivial. However, this is false: indeed, and this is part of the magic of the problem, the intersection is highly non-trivial, and it is not known—according to our current knowledge—whether a complete description of the intersection exists. Nevertheless, we can describe it partially: all n-fold commutators of every permutation of the n-tuple

$$\left\langle \left[\left\langle c_i^1 \right\rangle\right]_\sim \,\middle|\, i < n \right\rangle$$

are non-zero and lie in the intersection. For the general construction of n-fold commutators, see the appendix [A.2]. First, we present a method to compute the length of such n-fold commutator:

Lemma. *For $1 \leq k \leq n$, the length of a k-fold commutator of a k-tuple*

$$\left\langle \left[\left\langle c_{\sigma(i)}^1 \right\rangle\right]_\sim \,\middle|\, i < k \right\rangle \in F(\{c_i \mid i < n\})^k$$

for any injection $\sigma : k \hookrightarrow n$, where the bracketing follows a full binary rooted tree with k leaves $T \in \mathcal{T}_k$, satisfies:

$$\operatorname{length}\left(\left[T, \left\langle \left[\left\langle c_{\sigma(i)}^1 \right\rangle\right]_\sim \,\middle|\, i < k \right\rangle \circ \sigma\right]\right) = \sum_{s \in \operatorname{Leaves}(T)} 2^{|s|}$$

Proof. Since, for any wrapping words $W, V \in F(\{c_i \mid i < n\})$ such that they have representative words with letters (in $\{c_i \mid i < n\}$) that are disjoint, we have

$$\operatorname{length}([W, V]) \;=\; 2\operatorname{length}(W) + 2\operatorname{length}(V)\,.$$

It is convenient to proceed by induction (up until n) on the number of leaves $1 \leq k \leq n$ of the tree .

For the base case $k = 1$, there is only one such tree $\{\varepsilon\}$ with one leave ε. The corresponding commutator (for any $\sigma : 1 \hookrightarrow n$) is

$$\Big[\{\varepsilon\}, \Big\langle \Big[\Big\langle c^1_{\sigma(0)} \Big\rangle\Big]_\sim \Big\rangle\Big] = \Big[\Big\langle c^1_{\sigma(0)} \Big\rangle\Big]_\sim\,,$$

which has length $1 = 2^{|\varepsilon|}$. Now, for the inductive step, assume that the formula holds for all full binary rooted trees with j leaves where $1 \leq j \leq k < n$. We show that the formula holds for all full binary rooted trees with $k+1$ leaves. Let $T \in \mathcal{T}_{k+1}$. Since T is full and has at least 2 leaves. It then decomposes uniquely as the union of two full binary rooted trees

$$L_T \in \mathcal{T}_i \quad \text{and} \quad R_T \in \mathcal{T}_{k+1-i}$$

for

$$i := |\mathrm{Leaves}(L_T)| \in [\![1,\, k]\!],$$

whose leaves (which are order-isomorphic with $(\{0, \ldots, k\}, <)$) partition $\{0, \ldots, k\}$ into an initial block $\{0, \ldots, i-1\}$ and a final block $\{i, \ldots, k\}$. Then, by construction, (for any $\sigma : k+1 \hookrightarrow n$)

$$\begin{aligned} &\Big[T, \Big\langle \Big[\Big\langle c^1_{\sigma(j)} \Big\rangle\Big]_\sim \;\Big|\; j < k+1 \Big\rangle\Big] \\ &= \Big[\Big[L_T, \Big\langle \Big[\Big\langle c^1_{\sigma(j)} \Big\rangle\Big]_\sim \;\Big|\; j < i \Big\rangle\Big],\; \Big[R_T, \Big\langle \Big[\Big\langle c^1_{\sigma(j)} \Big\rangle\Big]_\sim \;\Big|\; i \leq j < k+1 \Big\rangle\Big]\Big]\,. \end{aligned}$$

Hence, by our earlier remark (as they trivially admit representatives with disjoint letters because σ is injective), the total length, $\operatorname{length}\Big(\Big[T, \Big\langle \Big[\Big\langle c^1_{\sigma(j)} \Big\rangle\Big]_\sim \;\Big|\; j < k+1 \Big\rangle\Big]\Big)$, is

$$2\operatorname{length}\Big(\Big[L_T, \Big\langle \Big[\Big\langle c^1_{\sigma(j)} \Big\rangle\Big]_\sim \;\Big|\; j < i \Big\rangle\Big]\Big) + 2\operatorname{length}\Big(\Big[R_T, \Big\langle \Big[\Big\langle c^1_{\sigma(j)} \Big\rangle\Big]_\sim \;\Big|\; i \leq j < k+1 \Big\rangle\Big]\Big)\,.$$

By the induction hypothesis applied to the i-tuple with injection $\sigma' := \sigma|_i : i \hookrightarrow n$ and bracketing following L_T,

$$\operatorname{length}\Big(\Big[L_T, \Big\langle \Big[\Big\langle c^1_{\sigma'(j)} \Big\rangle\Big]_\sim \;\Big|\; j < i \Big\rangle\Big]\Big) = \sum_{s_L \in \mathrm{Leaves}(L_T)} 2^{|s_L|}.$$

Similarly, by the induction hypothesis applied to the $k+1-i$-tuple with injection $\sigma'' := \sigma \circ (\cdot + i) : k+1-i \hookrightarrow n$ and bracketing following R_T,

$$\operatorname{length}\Big(\Big[R_T, \Big\langle \Big[\Big\langle c^1_{\sigma''(j)} \Big\rangle\Big]_\sim \;\Big|\; j < k+1-i \Big\rangle\Big]\Big) = \sum_{s_R \in \mathrm{Leaves}(R_T)} 2^{|s_R|}.$$

Thus,

$$\operatorname{length}\Big(\Big[T, \Big\langle \Big[\Big\langle c^1_{\sigma(j)} \Big\rangle\Big]_\sim \;\Big|\; j < k+1 \Big\rangle\Big]\Big) = \sum_{s_L \in \mathrm{Leaves}(L_T)} 2^{|s_L|+1} + \sum_{s_R \in \mathrm{Leaves}(R_T)} 2^{|s_R|+1}.$$

Since, by construction,

$$\mathrm{Leaves}(T) = \{\langle 0 \rangle \frown s_L \mid s_L \in \mathrm{Leaves}(L_T)\} \sqcup \{\langle 1 \rangle \frown s_R \mid s_R \in \mathrm{Leaves}(R_T)\}\,,$$

we obtain

$$\operatorname{length}\Big(\Big[T,\Big\langle\Big[\Big\langle c^1_{\sigma(j)}\Big\rangle\Big]_\sim\ \Big|\ j<k+1\Big\rangle\Big]\Big)=\sum_{s\in\mathrm{Leaves}(T)}2^{|s|},$$

as desired. This concludes the induction step. □

Remark. For any $\sigma\in\mathrm{Bij}(n)$ and a full binary rooted tree with n leaves $T\in\mathcal{T}_n$, let $L:=\operatorname{length}\Big(\Big[T,\Big\langle\Big[\Big\langle c^1_{\sigma(i)}\Big\rangle\Big]_\sim\ \Big|\ i<n\Big\rangle\Big]\Big)$ then:

- By the lemma, $n=\sum_{s\in\mathrm{Leaves}(T)}2^0\le L$.
- If T is comb with path $s\in\{0,1\}^{n-1}$, then clearly

  $$\mathrm{Leaves}(T)=\{s\}\sqcup\{s|_i\frown\langle 1-s(i)\rangle\mid i<n-1\},$$

  and we have, by the lemma,

  $$L=2^{n-1}+\sum_{i<n-1}2^{i+1}=2^{n-1}+2^n-2=3\cdot 2^{n-1}-2.$$

- If T is a subset of the perfect balanced full binary rooted tree $\{0,1\}^{\le\lceil\log_2(n)\rceil}$ then, by the lemma,

  $$L\le\sum_{i<2^{\lceil\log_2(n)\rceil}}2^{\lceil\log_2(n)\rceil}=2^{2\lceil\log_2(n)\rceil}\le 2^{2(\log_2(n)+1)}=4n^2.$$

  and in the special case of n being a power of 2, $\lceil\log_2(n)\rceil=\log_2(n)$ so $L=2^{2\log_2(n)}=n^2$.

Finally, we show that these n-fold commutators belong to $\mathcal{S}_n$.

Lemma. *Notation as in appendix [A.2], $\mathcal{S}_n$ contains*

$$\Big\{\Big[T,\Big\langle\Big[\Big\langle c^1_{\sigma(i)}\Big\rangle\Big]_\sim\ \Big|\ i<n\Big\rangle\Big]\ \Big|\ \sigma\in\mathrm{Bij}(n),\ T\in\mathcal{T}_n\Big\},$$

Proof. Fix $\sigma\in\mathrm{Bij}(n)$ and a full binary rooted tree with n leaves $T\in\mathcal{T}_n$. We show that

$$[\varepsilon]_\sim\ne\Big[T,\Big\langle\Big[\Big\langle c^1_{\sigma(i)}\Big\rangle\Big]_\sim\ \Big|\ i<n\Big\rangle\Big]\in\bigcap_{i<n}\ker\Big(\tilde\pi_{\{c_i\}}\Big).$$

First, since

$$\operatorname{length}\Big(\Big[T,\Big\langle\Big[\Big\langle c^1_{\sigma(i)}\Big\rangle\Big]_\sim\ \Big|\ i<n\Big\rangle\Big]\Big)\ge n>0=\operatorname{length}([\varepsilon]_\sim),$$

we must have

$$[\varepsilon]_\sim\ne\Big[T,\Big\langle\Big[\Big\langle c^1_{\sigma(i)}\Big\rangle\Big]_\sim\ \Big|\ i<n\Big\rangle\Big].$$

Second, to show that the n-fold commutator lies in the kernel, fix $j<n$. Then, because for any words $W,V\in F(\{c_i\mid i<n\})$ we have (as $\tilde\pi_{\{c_j\}}$ is a group morphism)

$$\tilde\pi_{\{c_j\}}([W,V])=\Big[\tilde\pi_{\{c_j\}}(W),\tilde\pi_{\{c_j\}}(V)\Big],$$

we must have

$$\tilde\pi_{\{c_j\}}\Big(\Big[T,\Big\langle\Big[\Big\langle c^1_{\sigma(i)}\Big\rangle\Big]_\sim\ \Big|\ i<n\Big\rangle\Big]\Big)=\Big[T,\Big\langle\tilde\pi_{\{c_j\}}\Big(\Big[\Big\langle c^1_{\sigma(i)}\Big\rangle\Big]_\sim\Big)\ \Big|\ i<n\Big\rangle\Big],$$

as stated in Property 1 of [A.2]. Now, since $\tilde{\pi}_{\{c_j\}}\left(\left[\left\langle c_j^1 \right\rangle\right]_\sim\right) = [\varepsilon]_\sim$, we have (because σ is a permutation)

$$[\varepsilon]_\sim \in \operatorname{ran}\left(\left\langle \tilde{\pi}_{\{c_j\}}\left(\left[\left\langle c_{\sigma(i)}^1 \right\rangle\right]_\sim\right) \,\middle|\, i < n \right\rangle\right),$$

and so

$$\left[T, \left\langle \tilde{\pi}_{\{c_j\}}\left(\left[\left\langle c_{\sigma(i)}^1 \right\rangle\right]_\sim\right) \,\middle|\, i < n \right\rangle\right] = [\varepsilon]_\sim,$$

as stated in Property 2 of [A.2]. This concludes since $j < n$ was arbitrary. □

To answer the problem, take any full binary rooted tree with n leaves, $T \in \mathcal{T}_n$, such that it is a subset of the perfect balanced full binary rooted tree $\{0,1\}^{\lceil \log_2(n) \rceil}$ (there exists at least one in any case, and there is only one when n is a power of 2). Then, as discussed, the wrapping word

$$W := \left[T, \left\langle \left[\left\langle c_i^1 \right\rangle\right]_\sim \,\middle|\, i < n \right\rangle\right] \in \mathcal{S}_n$$

and has length

$$\operatorname{length}(W) \leq 4n^2,$$

with the special case that $\operatorname{length}(W) = n^2$ when n is a power of 2.

- When $n = 2$, there is only one full binary rooted tree $T = \{\varepsilon, \langle 0 \rangle, \langle 1 \rangle\}$ with two leaves, which gives the wrapping word

  $$\left[T, \left\langle \left[\left\langle {c_0}^1 \right\rangle\right]_\sim, \left[\left\langle {c_1}^1 \right\rangle\right]_\sim \right\rangle\right] = \left[\left\langle {c_0}^{-1}, {c_1}^{-1}, {c_0}^1, {c_1}^1 \right\rangle\right]_\sim,$$

  i.e., the orient wrapping corresponding to it is:

  - 1 counterclockwise loop at c_0,
  - 1 counterclockwise loop at c_1,
  - 1 clockwise loop at c_0,
  - 1 clockwise loop at c_1.

- Similarly, for $n = 3$ the element of $\mathcal{T}_3$ are $T_1 = \{\varepsilon, \langle 0 \rangle, \langle 1 \rangle, \langle 0, 0 \rangle, \langle 0, 1 \rangle\}$ and $T_2 = \{\varepsilon, \langle 0 \rangle, \langle 1 \rangle, \langle 1, 0 \rangle, \langle 1, 1 \rangle\}$:

$$\left[T_1, \left\langle \left[\left\langle {c_0}^1 \right\rangle\right]_\sim, \left[\left\langle {c_1}^1 \right\rangle\right]_\sim, \left[\left\langle {c_2}^1 \right\rangle\right]_\sim \right\rangle\right] = \left[\left\langle {c_1}^{-1}, {c_0}^{-1}, {c_1}^1, {c_0}^1, {c_2}^{-1}, {c_0}^{-1}, {c_1}^{-1}, {c_0}^1, {c_1}^1, {c_2}^1 \right\rangle\right]_\sim,$$

$$\left[T_2, \left\langle \left[\left\langle {c_0}^1 \right\rangle\right]_\sim, \left[\left\langle {c_1}^1 \right\rangle\right]_\sim, \left[\left\langle {c_2}^1 \right\rangle\right]_\sim \right\rangle\right] = \left[\left\langle {c_0}^{-1}, {c_2}^{-1}, {c_1}^{-1}, {c_2}^1, {c_1}^1, {c_0}^1, {c_1}^{-1}, {c_2}^{-1}, {c_1}^1, {c_2}^1 \right\rangle\right]_\sim.$$

To illustrate the drastic economy in the length of the oriented wrapping that follows a full binary rooted tree $T \subseteq \{0,1\}^{\leq \lceil \log_2(n) \rceil}$, instead of one following a full binary rooted comb tree T_s (as one might have initially thought for the leftmost path $s = \underline{0}_{n-1} \in \{0,1\}^{n-1}$), we give examples of such a wrapping word in the free group without using the notation $[\cdot]_\sim$, $\langle \cdot \rangle$, the orientation $-1, 1$ and the concatenation symbol for simplicity.

Already, $4n^2 < 3 \cdot 2^{n-1} - 2$ if and only if $n \geq 8$, and $n^2 < 3 \cdot 2^{n-1} - 2$ if and only if $n \geq 3$. Thus, for the power of 2, $n = 4 = 2^2$, we have a wrapping word following $\{0,1\}^{\leq 2}$ of length $16 = 4^2$:

$$\begin{aligned}
\left[\{0,1\}^{\leq 2}, \langle c_0, c_1, c_2, c_3 \rangle\right] &= [[c_0, c_1], [c_2, c_3]] \\
&= [c_0, c_1]^{-1} [c_2, c_3]^{-1} [c_0, c_1] [c_2, c_3] \\
&= (c_0^{-1} c_1^{-1} c_0 c_1)^{-1} (c_2^{-1} c_3^{-1} c_2 c_3)^{-1} (c_0^{-1} c_1^{-1} c_0 c_1)(c_2^{-1} c_3^{-1} c_2 c_3) \\
&= c_1^{-1} c_0^{-1} c_1 c_0 \, c_3^{-1} c_2^{-1} c_3 c_2 \, c_0^{-1} c_1^{-1} c_0 c_1 \, c_2^{-1} c_3^{-1} c_2 c_3.
\end{aligned}$$

Meanwhile, following $T_{\underline{0}_3} = \{\varepsilon, \langle 1\rangle, \langle 0\rangle, \langle 0,0\rangle, \langle 0,1\rangle, \langle 0,0,0\rangle, \langle 0,0,1\rangle\}$ the wrapping word has length $22 = 3 \cdot 2^{4-1} - 2$:

$$\begin{aligned}
\left[T_{\underline{0}_3}, \langle c_0, c_1, c_2, c_3\rangle\right] &= \left[\left[T_{\underline{0}_2}, \langle c_0, c_1, c_2\rangle\right], c_3\right] \\
&= \left[T_{\underline{0}_2}, \langle c_0, c_1, c_2\rangle\right]^{-1} c_3^{-1} \left[T_{\underline{0}_2}, \langle c_0, c_1, c_2\rangle\right] c_3 \\
&= \left[\left[T_{\underline{0}_1}, \langle c_0, c_1\rangle\right], c_2\right]^{-1} c_3^{-1} \left[\left[T_{\underline{0}_1}, \langle c_0, c_1\rangle\right], c_2\right] c_3 \\
&= c_2^{-1} \left[T_{\underline{0}_1}, \langle c_0, c_1\rangle\right]^{-1} c_2 \left[T_{\underline{0}_1}, \langle c_0, c_1\rangle\right] c_3^{-1} \\
&\qquad\quad \left[T_{\underline{0}_1}, \langle c_0, c_1\rangle\right]^{-1} c_2^{-1} \left[T_{\underline{0}_1}, \langle c_0, c_1\rangle\right] c_2 c_3 \\
&= c_2^{-1} [c_0, c_1]^{-1} c_2 [c_0, c_1] c_3^{-1} [c_0, c_1]^{-1} c_2^{-1} [c_0, c_1] c_2 c_3 \\
&= c_2^{-1}c_1^{-1}c_0^{-1}c_1c_0c_2\, c_0^{-1}c_1^{-1}c_0c_1\, c_3^{-1}c_1^{-1}c_0^{-1}c_1c_0c_2^{-1}c_0^{-1}c_1^{-1}c_0c_1c_2c_3.
\end{aligned}$$

And, for example, on $8 = 2^3$ nails we have the wrapping following $\{0,1\}^{\leq 3}$ of length $64 = 8^2$, which is smaller than the one following $T_{\underline{0}_7}$ of length $382 = 3 \cdot 2^{8-1} - 2$. For completeness, we give them below:

$$\begin{aligned}
\left[\{0,1\}^{\leq 3}, \langle c_i \mid i < 8\rangle\right] = {}& c_3^{-1}c_2^{-1}c_3c_2\, c_1^{-1}c_0^{-1}c_1c_0\, c_2^{-1}c_3^{-1}c_2c_3\, c_0^{-1}c_1^{-1}c_0c_1 \\
& c_7^{-1}c_6^{-1}c_7c_6\, c_5^{-1}c_4^{-1}c_5c_4\, c_6^{-1}c_7^{-1}c_6c_7\, c_4^{-1}c_5^{-1}c_4c_5 \\
& c_1^{-1}c_0^{-1}c_1c_0\, c_3^{-1}c_2^{-1}c_3c_2\, c_0^{-1}c_1^{-1}c_0c_1\, c_2^{-1}c_3^{-1}c_2c_3 \\
& c_5^{-1}c_4^{-1}c_5c_4\, c_7^{-1}c_6^{-1}c_7c_6\, c_4^{-1}c_5^{-1}c_4c_5\, c_6^{-1}c_7^{-1}c_6c_7.
\end{aligned}$$

$$\left[T_{\underline{0}_7}, \langle c_i \mid i < 8\rangle\right] =$$

$$\begin{aligned}
& c_6c_5c_4c_3c_2c_0c_1c_0^{-1}c_1^{-1}c_2^{-1}c_1^{-1}c_0^{-1}c_1c_0c_3^{-1}c_0c_1c_0^{-1}c_1^{-1}c_2^{-1}c_1^{-1}c_0^{-1}c_1c_0c_2c_4^{-1}c_2c_0c_1c_0^{-1}c_1^{-1}c_2^{-1} \\
& c_1^{-1}c_0^{-1}c_1c_0c_3^{-1}c_0c_1c_0^{-1}c_1^{-1}c_2^{-1}c_1^{-1}c_0^{-1}c_1c_0c_2c_3c_5^{-1}c_3c_2c_0c_1c_0^{-1}c_1^{-1}c_2^{-1}c_1^{-1}c_0^{-1}c_1c_0c_3^{-1}c_0c_1c_0^{-1} \\
& c_1^{-1}c_2^{-1}c_1^{-1}c_0^{-1}c_1c_0c_2c_4^{-1}c_2c_0c_1c_0^{-1}c_1^{-1}c_2^{-1}c_1^{-1}c_0^{-1}c_1c_0c_3^{-1}c_0c_1c_0^{-1}c_1^{-1}c_2^{-1}c_1^{-1}c_0^{-1}c_1c_0c_2c_3c_4c_6^{-1} \\
& c_4c_3c_2c_0c_1c_0^{-1}c_1^{-1}c_2^{-1}c_1^{-1}c_0^{-1}c_1c_0c_3^{-1}c_0c_1c_0^{-1}c_1^{-1}c_2^{-1}c_1^{-1}c_0^{-1}c_1c_0c_2c_4^{-1}c_2c_0c_1c_0^{-1}c_1^{-1}c_2^{-1}c_1^{-1}c_0^{-1} \\
& c_1c_0c_3^{-1}c_0c_1c_0^{-1}c_1^{-1}c_2^{-1}c_1^{-1}c_0^{-1}c_1c_0c_2c_3c_5^{-1}c_3c_2c_0c_1c_0^{-1}c_1^{-1}c_2^{-1}c_1^{-1}c_0^{-1}c_1c_0c_3^{-1}c_0c_1c_0^{-1}c_1^{-1}c_2^{-1} \\
& c_1^{-1}c_0^{-1}c_1c_0c_2c_4^{-1}c_2c_0c_1c_0^{-1}c_1^{-1}c_2^{-1}c_1^{-1}c_0^{-1}c_1c_0c_3^{-1}c_0c_1c_0^{-1}c_1^{-1}c_2^{-1}c_1^{-1}c_0^{-1}c_1c_0c_2c_3c_4c_5^{-1}c_3c_2c_0c_1c_0^{-1} \\
& c_1^{-1}c_2^{-1}c_1^{-1}c_0^{-1}c_1c_0c_3^{-1}c_0c_1c_0^{-1}c_1^{-1}c_2^{-1}c_1^{-1}c_0^{-1}c_1c_0c_2c_4^{-1}c_2c_0c_1c_0^{-1}c_1^{-1}c_2^{-1}c_1^{-1}c_0^{-1}c_1c_0c_3^{-1}c_0c_1 \\
& c_0^{-1}c_1^{-1}c_2^{-1}c_1^{-1}c_0^{-1}c_1c_0c_2c_3c_4c_5c_6c_7
\end{aligned}$$

Remark. A generalization of the problem is the task of hanging a framed picture with a rope around a set of nails in such a way that it remains hanging on certain specified sets of nails, but falls if any of these sets are entirely removed (necessarily possibly along with additional nails). An instance of such a problem is the case where we take exactly all sets of size k, known as the k-out-of-n problem (we have already solved the 1-out-of-n problem). In the paper by Demaine et al. [1], they proved that all reasonable puzzles of this kind are solvable, and even more, by describing such problems using monotone Boolean function, and that for the k-out-of-n problem, the length of a solution can be bounded by a polynomial in n. They also discuss a connection with the Borromean rings. Other contributions or simplifications are also provided in the paper of Johan Wästlund [2].

A

A.1

Let $n \geq 1$ and let $\{c_i \mid i < n\}$ be a finite set of size n. We define formally in set theory (**ZF**) the free group generated by $\{c_i \,|\, i < n\}$, $F\left(\{c_i \,|\, i < n\}\right)$, as follows. First, we introduce an orientation by defining $c_i^1 := (c_i, 1)$ and $c_i^{-1} := (c_i, -1)$. Then, we consider the set of all possible words over the alphabet $C_n := \left\{c_i^1 \,\middle|\, i \in n\right\} \cup \left\{c_i^{-1} \,\middle|\, i \in n\right\}$, that is, the set of all sequences w with domain $\mathrm{dom}(w) \in \mathbb{N}$ and range $\mathrm{ran}(w) \subset C_n$:

$$C_n^{<\omega} = \bigcup_{i \in \mathbb{N}} C_n^i.$$

We let the binary operation of concatenation $\frown\colon C_n^{<\omega} \times C_n^{<\omega} \to C_n^{<\omega}$ of two words $w, v \in C_n^{<\omega}$ defined by $w \frown v : \mathrm{dom}(w) + \mathrm{dom}(v) \to C_n$ where for $i \in \mathrm{dom}(w) + \mathrm{dom}(v)$:

$$(w \frown v)(i) = \begin{cases} w_i & \text{if } i < \mathrm{dom}(w), \\ v_j & \text{if } \exists j \geq 0 \quad i = \mathrm{dom}(w) + j. \end{cases}$$

The reader may check that concatenation is internal, associative, admits a neutral element (namely the empty sequence $\varepsilon = \varnothing$), and that it is non-commutative as long as $n \geq 2$. So that $\langle C_n^{<\omega}, \frown, \varepsilon \rangle$ is a monoïd. Then, we let the smallest equivalence relation $\sim \subset C_n^{<\omega} \times C_n^{<\omega}$ containing the set:

$$C\left(n\right) := \left\{\left(\left\langle c_i^1, c_i^{-1}\right\rangle, \varepsilon\right) \,\middle|\, i < n\right\} \cup \left\{\left(\left\langle c_i^{-1}, c_i^1\right\rangle, \varepsilon\right) \,\middle|\, i < n\right\},$$

such that the operation of concatenation passes to the quotient:

$$\sim := \bigcap \left\{ R \in \mathscr{P}\left(C_n^{<\omega} \times C_n^{<\omega}\right) \,\middle|\, \begin{array}{l} \text{“}R \text{ is an equivalence relation”} \wedge C\left(n\right) \subset R \\ \wedge \quad \forall v, w, v', w' \in C_n^{<\omega} \\ \qquad (v, v') \in R \wedge (w, w') \in R \to (v \frown w, v' \frown w') \in R \end{array} \right\}.$$

This set of such equivalence relations is non-empty, as it contains the trivial equivalence relation $\mathscr{P}\left(C_n^{<\omega} \times C_n^{<\omega}\right)$ where everything is equivalent to everything, so the intersection is not over the empty set and is thus well-defined. It is easy to see that an arbitrary intersection of equivalence relations is an equivalence relation. For simplicity, we define for any $w \in C_n^{<\omega}$ and $l \geq 0$ the abbreviation:

$$w^l := \frown_{s<l} w$$

(the order of concatenation does not matter here). For example, we have $w^0 = \varepsilon$, since we perform a concatenation with an index running over the empty set, which is then the empty set. Every equivalent word will behave the same under the "new" concatenation by the construction of $\sim$. In particular, for any words $w, v \in C_n^{<\omega}$, any $i < n$, any $\theta \in \{1, -1\}$, and any $r, s > 0$ we have that:

$$\text{if } r \geq s: \quad \left(w \frown \left\langle c_i^\theta \right\rangle^r \frown \left\langle c_i^{-\theta} \right\rangle^s \frown v\right) \sim w \frown \left\langle c_i^\theta \right\rangle^{r-s} \frown v,$$

while:

$$\text{if } r < s: \quad \left(w \frown \left\langle c_i^\theta \right\rangle^r \frown \left\langle c_i^{-\theta} \right\rangle^s \frown v\right) \sim w \frown \left\langle c_i^{-\theta} \right\rangle^{s-r} \frown v,$$

For example:

$$\left\langle c_0^1 \right\rangle^3 \frown \left\langle c_0^{-1} \right\rangle^4 = \left\langle c_0^1, c_0^1, c_0^1, c_0^{-1}, c_0^{-1}, c_0^{-1}, c_0^{-1} \right\rangle \sim \left\langle c_0^{-1} \right\rangle.$$

With this, it is easy to fully characterise the equivalence relation $\sim$: two words $w, w' \in C_n^{<\omega}$ are equivalent if and only if, after recursively removing every occurrence of cancelling adjacent pairs (for $i < n$ and $\theta \in \{-1, 1\}$)

$$\left\langle c_i^{\theta}, c_i^{-\theta} \right\rangle$$

from w and w' to obtain "clean"[3] words v and v', the resulting words are equal, $v = v'$.

Indeed, for the "only if" condition, we notice that the described process is clearly reflexive, symmetric, and transitive, and thus defines an equivalence relation R on $C_n^{<\omega}$. The relation R trivially contains $C(n)$. To see that concatenation passes to the quotient under this relation, let $(w, w'), (w'', w''') \in R$, and let the respective "clean" words be v, v', v'', v'''. We have $v = v'$ and $v'' = v'''$. Now, the "clean" words $\overline{v}$ and $\overline{\overline{v}}$ of $w \frown w''$ and $w' \frown w'''$ are respectively exactly the clean words of $v \frown v''$ and $v' \frown v'''$, because, regardless of the order of execution of the cleaning, we always obtain the same clean word in the end. Moreover, $v \frown v'' = v' \frown v'''$, so because cleaning completely a word is deterministic, the corresponding clean words must be equal. Hence, $(w \frown w'', w' \frown w''') \in R$ and R passes to the quotient. By construction, we thus obtain $\sim \subset R$ which concludes.

For the "if" condition, one obtains $w \sim v$ and $w' \sim v'$ (by the above remark and by recursive application of transitivity and, possibly, reflexivity). Thus, $v' \sim w'$ by symmetry, and hence, if $v = v'$, then $w \sim w'$ by transitivity.

After having constructed $\sim$, we can define a 2-ary relation ${}^\frown \subset (C_n^{<\omega}/_\sim \times C_n^{<\omega}/_\sim) \times C_n^{<\omega}/_\sim$ by:

$${}^\frown := \left\{ \left(\left([v]_\sim , [w]_\sim \right), [v \frown w]_\sim \right) \in \left(C_n^{<\omega}/_\sim \times C_n^{<\omega}/_\sim \right) \times C_n^{<\omega}/_\sim \,\middle|\, \exists v, w \in C_n^{<\omega} \right\}.$$

The construction of $\sim$ gives us that in fact ${}^\frown$ is a functional relation with domain $C_n^{<\omega}/_\sim \times C_n^{<\omega}/_\sim$:

$${}^\frown : C_n^{<\omega}/_\sim \times C_n^{<\omega}/_\sim \to C_n^{<\omega}/_\sim$$

satisfying:

$$[v]_\sim {}^\frown [w]_\sim = [v \frown w]_\sim .$$

(That is, ${}^\frown$ commutes with $\frown$ through the quotient map.) Now, the reader can verify that this new binary function ${}^\frown$ makes

$$F\left(\{ c_j \,|\, j < n \} \right) := C_n^{<\omega}/_\sim$$

into a group: its construction forces ${}^\frown$ to inherit the properties of $\frown$: closure, associativity, and the existence of a neutral element (here $[\varepsilon]_\sim$). A left and right inverse element of the equivalence class of a word $w \in C_n^{<\omega}$ exists, since it is easily seen to be the equivalence class of the word obtained by reversing the order of w and inverting each exponent (changing 1 to -1 and vice versa). More formally, if we denote by

$$p_2 : \{ c_i \,|\, i < n \} \times \{1, -1\} \to \{1, -1\}$$

[3] Technically, we define a cleaning function Clean : $C_n^{<\omega} \to C_n^{<\omega}$ that is the identity if the word has no cancelling adjacent pair; otherwise, it removes the adjacent pair at the far left of the word (the minimum coordinate). Observe that each time such a pair is removed, a new cancelling adjacent pair may appear. For each word w, the function $f_w : \mathbb{N} \to \mathbb{N}, \quad n \mapsto \mathrm{dom} \circ \mathrm{Clean}^{\circ n}(w)$ is decreasing and, since $\mathbb{N}$ is discrete and bounded below, it must stabilize: there exists an integer $m \geq 0$ such that for all $n \geq m$, $f_w(m) = f_w(n)$. The smallest such m is the number of steps required to clean the word w. The number m of steps is independent of other choices of the location in the definition Clean function. The "clean" word obtain from w is defined to be $\mathrm{Clean}^{\circ m}(w)$ and is independant of the choices of the location in the definition of the Clean function.

the projection onto the second coordinate, then, denoting the reversed word by

$$w^* := \frown_{i\in\mathrm{dom}(w)} \left\langle w_{\mathrm{dom}(w)-1-i}^{-p_2\left(w_{\mathrm{dom}(w)-1-i}\right)} \right\rangle ,$$

the inverse is $[w]_\sim^{-1} = [w^*]_\sim$. For example, we have

$$\left[\left\langle c_0^1 \right\rangle\right]_\sim^{-1} = \left[\left\langle c_0^1 \right\rangle^*\right]_\sim = \left[\left\langle c_0^{-1} \right\rangle\right]_\sim ,$$

which is consistent with $\langle c_0^1 \rangle \frown \left\langle c_0^{-1} \right\rangle \sim \varepsilon$. Moreover, we have the property that for any word $w \in C_n^{<\omega}$ and $l \in \mathbb{Z}$:

$$[w]_\sim^l = \begin{cases} \left[w^l\right]_\sim , & \text{if } l \geq 0, \\ \left[(w^*)^{|l|}\right]_\sim , & \text{if } l < 0. \end{cases}$$

One can check that $\frown$ is again non-commutative as long as $n \geq 2$ (since $c_0 \neq c_1$).

- $F(\{c_i \mid i < n\})$ has the following *free* property: for any group $\langle G, \cdot, e\rangle$ and any function $f : \{c_i \mid i < n\} \to G$, there exists a unique group homomorphism $\tilde{f} : F(\{c_i \mid i < n\}) \to G$ "extending" f, i.e. for any $i < n$, we have $\tilde{f}\left([\langle c_i^1 \rangle]_\sim\right) = f(c_i)$. Indeed, we can define a well-defined function $\overline{f} : C_n^{<\omega} \to G$; for $w \in C_n^{<\omega}$

$$\overline{f}(w) = \prod_{i\in\mathrm{dom}(w)} f(p_1(w_i))^{p_2(w_i)} ,$$

where the product is w.r.t to the internal operation $\cdot$ of G and taken in the order of the indices in $\mathrm{dom}(w)$ while p_1 denotes the projection onto the first coordinate. $\overline{f}$ is clearly a monoïd morphism. Moreover, with a simple mental exercise, it is the unique monoïd morphism for the property: for any $i < n$ and $\theta \in \{1, -1\}$, $\overline{f}\left(\left\langle c_i^\theta \right\rangle\right) = f(c_i)^\theta$. Now, fix a word $w \in C_n^{<\omega}$, since $\overline{f}$ is a monoid morphism, each cancelling adjacent pair in w: $\left\langle c_i^\theta, c_i^{-\theta} \right\rangle$ is sent to e (by the property of $\overline{f}$). Hence, removing a single cancelling adjacent pair from w to obtain a word w' does not change its image: $\overline{f}(w) = \overline{f}(w')$. By induction, we obtain that for each cleaning step of the word w, the newly obtained word has the same image as w. In particular, when we reach its clean version v (which is equivalently defined as the word with smallest domain among the representatives of $[w]_\sim$), we have $\overline{f}(w) = \overline{f}(v)$. Hence, if $w, w' \in C_n^{<\omega}$ are such that $w \sim w'$, and we denote their clean versions by $v, v' \in C_n^{<\omega}$, we have, as discussed, $\overline{f}(w) = \overline{f}(v)$ and $\overline{f}(w') = \overline{f}(v')$. Because $w \sim w'$, we know by our characterisation of the equivalence relation that $v = v'$, so $\overline{f}(v) = \overline{f}(v')$. By transitivity of equality, we then obtain $\overline{f}(w) = \overline{f}(w')$. Hence, $\overline{f}$ passes to the quotient. By the quotient property, there exists a unique function $\tilde{f} : F(\{c_i \mid i < n\}) \to G$ such that $\tilde{f} \circ q = \overline{f}$, where $q : C_n^{<\omega} \to F(\{c_i \mid i < n\})$ is the quotient map. Clearly, $\tilde{f}$ is a group homomorphism and is "extending" f. One easily shows that it is the unique group homomorphism $F(\{c_i \mid i < n\}) \to G$ "extending" f: any such group homomorphism g must be such that $g \circ q$ is a monoïd morphism satisfying the same property as $\overline{f}$ do. Hence, by unicity of $\overline{f}$, we have $g \circ q = \overline{f}$, and thus by the uniqueness in the quotient property, $g = \tilde{f}$.

Remark. In fact, if we replace $\{c_i \,|\, i < n\}$ with an arbitrary set J, we obtain—*mutatis mutandis*—the construction of the free group $F(J)$ generated by J (valid even if $J = \varnothing$). When working on $F(J)$, especially when writing its words, we usually omit for simplicity the pedantic notation of $[\_]_\sim$ and $\langle \_ \rangle$. For more information about this group, you can check the following Wikipedia page.

A.2

For a set Σ, let $\Sigma^* = \Sigma^{<\omega}$ be the set of finite words (including the empty word) over Σ. For $u, v \in \Sigma^*$, write $u \preceq v$ iff u is a prefix of v, i.e., $\exists w \in \Sigma^*$ such that $v = u \frown w$. Clearly, $(\Sigma^*, \preceq)$ is a poset. A subset $T \subseteq \Sigma^*$ is a *tree* if it is finite and prefix-closed: if $v \in T$ and $u \preceq v$, then $u \in T$. We say T is *rooted* if $\varepsilon \in T$[4]. If $|\Sigma| = 2$, the adjective *binary* is added and it is a *binary tree.* We say it is *full* if Σ is finite and every element is either a leaf (that is, a $\preceq|_{T\times T}$-maximal element in the subposet $(T, \preceq|_{T\times T})$) or has exactly all $|\Sigma|$ possible children in T:

$$\forall s \in T:\ (\forall \sigma \in \Sigma,\ s \frown \langle\sigma\rangle \in T) \quad \text{or} \quad (\forall \sigma \in \Sigma,\ s \frown \langle\sigma\rangle \notin T)\,.$$

The set of leaves of a tree T (that is, $\preceq|_{T\times T}$-maximal elements)[5] is, if T is a full rooted tree, also equal to

$$\mathrm{Leaves}(T) = \{\, s \in T \mid \forall \sigma \in \Sigma,\ s \frown \langle\sigma\rangle \notin T \,\}\,.$$

Now, fix $\Sigma = \{0, 1\}$ and $n \geq 1$, define

$$\mathcal{T}_n := \{\, T \subseteq \Sigma^* \mid \text{"}T \text{ is a full binary rooted tree" and } |\mathrm{Leaves}(T)| = n \,\}\,.$$

Equip Σ^* with the lexicographic order induced by $0 < 1$. For $T \in \mathcal{T}_n$ the leaves are then naturally ordered $(\mathrm{Leaves}(T), \prec_{\mathrm{lex}}|_{\mathrm{Leaves}(T)\times\mathrm{Leaves}(T)})$ and can thus be identified unambiguously to the ordered set $(\{0, ..., n-1\}, <)$:

$$\ell_0 \prec_{\mathrm{lex}} \cdots \prec_{\mathrm{lex}} \ell_{n-1}.$$

For example ($n = 3$), the element of $\mathcal{T}_3$ are $\{\varepsilon,\ \langle 0\rangle,\ \langle 1\rangle,\ \langle 0, 0\rangle,\ \langle 0, 1\rangle\}$ and $\{\varepsilon,\ \langle 0\rangle,\ \langle 1\rangle,\ \langle 1, 0\rangle,\ \langle 1, 1\rangle\}$. Here the leaves are, in lexicographic order $\langle 0, 0\rangle \prec \langle 0, 1\rangle \prec \langle 1\rangle$ and $\langle 0\rangle \prec \langle 1, 0\rangle \prec \langle 1, 1\rangle$ respectively.

Remark. $\mathcal{T}_1 = \{\{\varepsilon\}\}$, and if $n \geq 1$ then $\bigsqcup_{i=1}^{n} (\mathcal{T}_i \times \mathcal{T}_{n+1-i}) \simeq \mathcal{T}_{n+1}$ [6]. We conclude by induction that $\mathcal{T}_n$ is finite and satisfies $|\mathcal{T}_1| = 1$, with

$$|\mathcal{T}_{n+1}| = \sum_{i=1}^{n} |\mathcal{T}_{n+1-i}| \cdot |\mathcal{T}_i|\,,$$

i.e. $|\mathcal{T}_n| = C_{n-1}$, the $(n-1)$-st Catalan number (because they are defined by this recurrence relation).

For each full binary rooted tree $T \subset \Sigma^*$, define its height by

$$l(T) := \max\{|s| \mid s \in T\}\,,$$

We have the following lemma:

[4] Since the empty word is a prefix of any word, we could have equivalently replaced the condition by $T \neq \varnothing$.

[5] This set is non-empty as soon as T is non-empty, since T is finite and hence a $\preceq|_{T\times T}$-maximal element must exist.

[6] The bijection is defined by $f : (L, R) \mapsto \{\varepsilon\} \cup \{\, \langle 0\rangle \frown s \mid s \in L \,\} \cup \{\, \langle 1\rangle \frown t \mid t \in R \,\}$. This is clearly a full binary rooted tree with the leaves

$$\{\, \langle 0\rangle \frown \ell \mid \ell \in \mathrm{Leaves}(L) \,\} \sqcup \{\, \langle 1\rangle \frown r \mid r \in \mathrm{Leaves}(R) \,\}\,,$$

so has exactly $|\mathrm{Leaves}(L)| + |\mathrm{Leaves}(R)| = i + (n + 1 - i) = n + 1$. Thus the map is well defined and is easily seen to be injective. An inverse of this map is constructed as follows. Since $|\mathrm{Leaves}(T)| = n + 1 \geq 2$, the root ε cannot be a leaf, so both children $\langle 0\rangle$ and $\langle 1\rangle$ belong to T. We define

$$L_T := \{\, s \in \Sigma^* \mid \langle 0\rangle \frown s \in T \,\}\,, \qquad R_T := \{\, t \in \Sigma^* \mid \langle 1\rangle \frown t \in T \,\}\,.$$

Then L_T and R_T are prefix-closed finite subsets of Σ^* containing ε, and they inherit the full binary property from T. Hence, for $i := |\mathrm{Leaves}(L_T)|$, we must have $1 \leq i \leq n$ and $L_T \in \mathcal{T}_i$ and $R_T \in \mathcal{T}_{n+1-i}$. Clearly the map $g : T \mapsto (L_T, R_T)$ is a right inverse of f; hence f is surjective and therefore bijective with $f^{-1} = g$.

Lemma. *For a full binary rooted tree $T \subset \Sigma^*$, one has*

$$2^{|\text{Leaves}(T)|-1} \;\geq\; 2^{l(T)} \;\geq\; |\text{Leaves}(T)|.$$

The lower bound is tight for the sole perfect balanced full binary rooted tree of height h namely $\{0,1\}^{\leq h}$. The upper bound is tight for all of the 2^h full binary rooted comb[7] *trees of height h.*

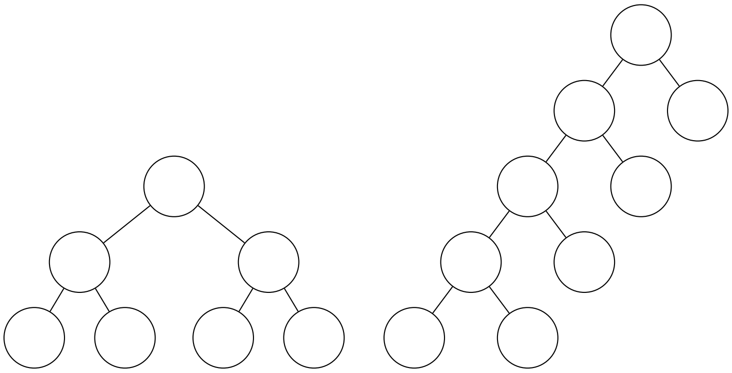

Figure 5: Left: Perfect balanced full binary rooted tree of height 2. Right: Full binary rooted comb tree of height 4.

Proof. We prove this by induction on the height $l(T)$ of the tree. For the base case, $l(T) = 0$ is equivalent to $T = \{\varepsilon\}$, so $|\text{Leaves}(T)| = 1 = 2^0 = 2^{1-1}$, and the statement holds.

Now, for the induction step, assume the statement holds for all full binary rooted trees of height at most $h \geq 0$. Let T be a full binary tree of height $l(T) = h + 1$. Since T is full and has height at least 1, it has at least 2 leaves. It then decomposes uniquely as the union of two full binary rooted trees $L_T \in \mathcal{T}_i$ and $R_T \in \mathcal{T}_{|\text{Leaves}(T)|-i}$ for

$$i := |\text{Leaves}(L_T)| \in [\![1,\, |\text{Leaves}(T)| - 1]\!],$$

(the map f^{-1} in the previous footnote) whose leaves (which are order-isomorphic with $(\{0, \dots, |\text{Leaves}(T)| - 1\}, <))$ partition $\{0, \dots, |\text{Leaves}(T)| - 1\}$ into an initial block $\{0, \dots, i - 1\}$ and a final block $\{i, \dots, |\text{Leaves}(T)| - 1\}$.

By construction, $l(L_T),\, l(R_T) \;\leq\; \max\{l(L_T),\, l(R_T)\} \;=\; l(T) - 1 \;=\; h$. Hence, by the induction hypothesis,

$$|\text{Leaves}(L_T)| \leq 2^{l(L_T)} \leq 2^{|\text{Leaves}(L_T)|-1}, \qquad |\text{Leaves}(R_T)| \leq 2^{l(R_T)} \leq 2^{|\text{Leaves}(R_T)|-1}.$$

Therefore,

$$|\text{Leaves}(T)| = |\text{Leaves}(L_T)| + |\text{Leaves}(R_T)| \leq 2^{l(L_T)} + 2^{l(R_T)} \leq 2^h + 2^h = 2^{h+1} = 2^{l(T)}.$$

Moreover, since $|\text{Leaves}(L_T)|$, $|\text{Leaves}(R_T)| \leq |\text{Leaves}(T)| - 1$,

$$2^{l(T)} = 2^{\max\{l(L_T),\, l(R_T)\}+1} \leq \max\Big\{2^{|\text{Leaves}(L_T)|},\, 2^{|\text{Leaves}(R_T)|}\Big\} \leq 2^{|\text{Leaves}(T)|-1}.$$

Thus, in total,

$$|\text{Leaves}(T)| \leq 2^{l(T)} \leq 2^{|\text{Leaves}(T)|-1},$$

[7] For $s \in \Sigma^h$, a tree $T \subset \Sigma^*$ of height h is comb with path s if $\forall s' \in T,\ s' \preceq s \ \vee\ s' \in \text{Leaves}(T)$. In the case $|\Sigma| = 2$, it is easy to see that any comb tree with path s is a subset of the full binary rooted comb tree with path s, T_s.

which concludes the induction step. Clearly for $h \geq 0$, $\{0,1\}^{\leq h}$ has 2^h leaves, while for any $s \in \{0,1\}^h$, the full binary rooted comb tree T_s of height h has $h+1$ leaves (easily proven by induction). □

Now, let $\langle G, \cdot, e\rangle$ be a group. We present the construction of the set of any k-fold commutators (for $k \geq 1$), that is, all possible bracketings of the commutator for any k elements of G. The important point is that elements of $\mathcal{T}_k$ can be used to parametrise all possible binary bracketings of k entries.

For the sake of generality, let any binary map $\mathcal{B} : G \times G \longrightarrow G$ (we can apply the construction to $[\_,\_]$ afterwards). We define recursively on $\mathbb{N}_{>0}$ an evaluation map

$$\mathrm{eval}_{\mathcal{B}} : \bigsqcup_{k\geq 1} \left(\mathcal{T}_k \times G^k\right) \longrightarrow G$$

as follows. For $k = 1$, let any $\underline{g} \in G^k$, and set

$$\mathrm{eval}_{\mathcal{B}}((\{\varepsilon\}, \underline{g})) := g_0.$$

Now, for $k \geq 2$, assume we have constructed $\mathrm{eval}_{\mathcal{B}}$ on $\bigsqcup_{1\leq j\leq k-1} \left(\mathcal{T}_j \times G^j\right)$. We extend the map to $\bigsqcup_{1\leq j\leq k} \left(\mathcal{T}_j \times G^j\right)$. Because $k \geq 2$, any $T \in \mathcal{T}_k$ decomposes uniquely as the union of two full binary rooted trees $L_T \in \mathcal{T}_i$ and $R_T \in \mathcal{T}_{k-i}$ for $i := |\mathrm{Leaves}(L_T)| \in [\![1, k-1]\!]$, whose leaves (which are order-isomorphic with $(\{0,\ldots,k-1\},<)$) partition $\{0,\ldots,k-1\}$ into an initial block $\{0,\ldots,i-1\}$ and a final block $\{i,\ldots,k-1\}$. For any $\underline{g} \in G^k$, let $\underline{g}' := \underline{g}|_{[\![0,i-1]\!]} \in G^i$ and $\underline{g}'' := \underline{g}|_{[\![i,k-1]\!]} \circ (\cdot - i) \in G^{k-i}$; then $\underline{g} = \underline{g}' \frown \underline{g}''$. Furthermore, $\mathrm{eval}_{\mathcal{B}}((L_T, \underline{g}')), \mathrm{eval}_{\mathcal{B}}((R_T, \underline{g}'')) \in G$ are well defined by the recursion hypothesis, and we set

$$\mathrm{eval}_{\mathcal{B}}((T, \underline{g})) := \mathcal{B}\Big(\mathrm{eval}_{\mathcal{B}}((L_T, \underline{g}')),\, \mathrm{eval}_{\mathcal{B}}((R_T, \underline{g}''))\Big)\,.$$

By construction, this produces exactly the group element obtained by applying the binary map $\mathcal{B}$ according to the bracketing encoded by T. For example ($k = 3$), the set $\mathcal{T}_3$ has two trees (given before), which yield the two bracketings (for $\underline{g} = \langle g_1, g_2, g_3\rangle \in G^3$)

$$\mathcal{B}((\mathcal{B}((g_1, g_2)), g_3)) \qquad \text{and} \qquad \mathcal{B}((g_1, \mathcal{B}((g_2, g_3)))) \qquad \text{respectively.}$$

Notice the two properties:

1. If $f : G \to G$ is a function such that

$$f(\mathcal{B}((g_1, g_2))) = \mathcal{B}((f(g_1), f(g_2)))\,,$$

then, by induction, for any $T \in \mathcal{T}_k$ and $\underline{g} \in G^k$ with $k \geq 1$, we have

$$f\Big(\mathrm{eval}_{\mathcal{B}}\Big((T, \underline{g})\Big)\Big) = \mathrm{eval}_{\mathcal{B}}\Big((T, f \odot \underline{g})\Big)\,,$$

where $f \odot \underline{g}$ denotes the k-tuple obtained by applying f to each coordinate of $\underline{g}$.

2. If, for any $g \in G$,

$$\mathcal{B}((e, g)) = e = \mathcal{B}((g, e))\,,$$

(a property which is satisfied by $[\_,\_]$) then, by induction, for any $T \in \mathcal{T}_k$ and $\underline{g} \in G^k$ with $k \geq 2$, if $e \in \mathrm{ran}(\underline{g})$, then

$$\mathrm{eval}_{\mathcal{B}}\Big((T, \underline{g})\Big) = e.$$

- For simplicity, for $(T,\underline{g}) \in \bigsqcup_{k\geq 1}\Big(\mathcal{T}_k \times G^k\Big)$, we denote $\Big[T,\underline{g}\Big] := \mathrm{eval}_{[\_,\_]}\Big((T,\underline{g})\Big)$.

Remark. The tree viewpoint is convenient to enumerate all bracketings without ambiguity. Typically, different trees $T \in \mathcal{T}_n$ and different permutations of entries give different elements. For example, take the group $(\mathbb{R}_{>0},\cdot,1)$ and $\mathcal{B} : \mathbb{R}_{>0} \times \mathbb{R}_{>0} \to \mathbb{R}_{>0}$ defined by $\mathcal{B}(a,b) := a^b$. Then already in $\mathcal{T}_3$ the two bracketings are different, simply because $(a^b)^c = a^{bc} \neq a^{(b^c)}$ in general: for $\langle a,b,c\rangle = \langle 2,1,3\rangle$, the left-hand side is 2^3, while the right-hand side is 2. Now, if we permute the entries to obtain $\langle 1,3,2\rangle$, one has for the tree $T = \{\varepsilon, \langle 0\rangle, \langle 1\rangle, \langle 0,0\rangle, \langle 0,1\rangle\} \in \mathcal{T}_3$, that

$$\mathrm{eval}_{\mathcal{B}}\big((T,\langle 1,3,2\rangle)\big) = 1 \neq 2 = \mathrm{eval}_{\mathcal{B}}\big((T,\langle 2,1,3\rangle)\big).$$

References

[1] Erik D. Demaine et al. "Picture-Hanging Puzzles". In: *Theory of Computing Systems* 54.4 (2014). Journal version of FUN 2012 paper, pp. 531–550. DOI: 10.1007/s00224-013-9501-0. arXiv: 1203.3602 [cs.DS]. URL: https://arxiv.org/abs/1203.3602.

[2] Johan Wästlund. *Faulty picture-hanging improved.* 2021. DOI: 10.48550/arXiv.2102.00984. arXiv: 2102.00984 [cs.DM]. URL: https://arxiv.org/pdf/2102.00984.