Choiceless Computation and Symmetry: Limitations of Definability

Benedikt Pago
Mathematical Foundations of Computer Science, RWTH Aachen University, Germany
pago@logic.rwth-aachen.de

Abstract

The search for a logic capturing PTIME is a long standing open problem in finite model theory. One of the most promising candidate logics for this is *Choiceless Polynomial Time* with counting (CPT). Abstractly speaking, CPT is an isomorphism-invariant computation model working with hereditarily finite sets as data structures.

While it is easy to check that the evaluation of CPT-sentences is possible in polynomial time, the converse has been open for more than 20 years: Can every PTIME-decidable property of finite structures be expressed in CPT?

We attempt to make progress towards a negative answer and show that Choiceless Polynomial Time cannot compute a *preorder* with colour classes of *logarithmic size* in every hypercube. The reason is that such preorders have super-polynomially many automorphic images, which makes it impossible for CPT to define them.

While the computation of such a preorder is not a decision problem that would immediately separate P and CPT, it is significant for the following reason: The so-called Cai-Fürer-Immerman (CFI) problem is one of the standard "benchmarks" for logics and maybe best known for separating fixed-point logic with counting (FPC) from P. Hence, it is natural to consider this also a potential candidate for the separation of CPT and P. The strongest known positive result in this regard says that CPT is able to solve CFI if a preorder with logarithmically sized colour classes is present in the input structure.

Our result implies that this approach cannot be generalised to unordered inputs. In other words, CFI on unordered hypercubes is a PTIME-problem which provably cannot be tackled with the state-of-the-art choiceless algorithmic techniques.

2012 ACM Subject Classification Theory of computation → Finite Model Theory; Mathematics of computing → Permutations and combinations

Keywords and phrases finite model theory, descriptive complexity, choiceless computation, symmetries of combinatorial objects

Digital Object Identifier 10.4230/LIPIcs.CSL.2021.33

Acknowledgements I would like to thank my advisor Erich Grädel for helpful comments and discussions.

1 Introduction

One of the big open questions in descriptive complexity theory is whether there exists a logic capturing PTIME (see [4], [10], [11], [13]). Towards an answer to this question, several logics of increasing expressive power within PTIME have been devised, the best-studied of which is probably FPC, *fixed-point logic with counting* (see [5] for a survey). However, FPC only corresponds to a strict subset of PTIME because it cannot express the so-called *CFI query*, a version of the graph isomorphism problem on certain graphs constructed by Cai, Fürer and Immerman in 1992 [3]. This problem is in P and has turned out to be extremely valuable as a benchmark for PTIME-logics as well as for certain classes of graph isomorphism algorithms.

The most important candidate logics for capturing PTIME, which have not yet fallen prey to the CFI problem, are *Rank logic* [6] and *Choiceless Polynomial Time* (CPT). CPT was introduced in 1999 by Blass, Gurevich and Shelah [2] as a machine model that comes as close to Turing machines as possible, while enforcing *isomorphism-invariance* of the computations – this property is precisely the main difference between logics and classical Turing machines. Since its original invention, various different formalisations of CPT have emerged but the underlying principle is always the same: *Symmetric* computation on *polynomially-sized hereditarily finite sets* as data structures.

Not many lower bound results for Choiceless Polynomial Time are known so far, and of course, no *decision problem* in P has been shown to be undefinable in CPT. However, what has been achieved is a non-definability statement for a *functional problem*: Rossman showed that CPT cannot define the dual of any given finite vector space [15]. Our contribution is a result of a similar kind, but stronger in a sense: We show non-definability not only for a concrete functionally determined object, but for all objects satisfying a certain set of properties. Concretely, no CPT program can define a hereditarily finite set representing a preorder with colour classes of logarithmic size in every hypercube. This can be seen – potentially – as a first step towards a non-definability result for a decision problem: the already mentioned CFI query; this would separate CPT from PTIME. Let us explain what undefinable preorders have to do with the CFI problem (see Section 3 for details).

Recall that a preorder in a structure can be seen as a linear order on a collection of *colour classes*, which form a partition of the universe: These colour classes are subsets of the structure whose elements are pairwise indistinguishable. The smaller the colour classes are, the "finer" is the preorder, and the more closely it resembles a linear order. By the famous Immerman-Vardi Theorem ([12], [17]), fixed-point logic, and therefore also CPT, captures PTIME on linearly ordered structures. Therefore, intuitively speaking, hard problems like CFI should become easier to handle if CPT is able to define a sufficiently fine preorder, or even a linear order, on the input structure. Indeed, Pakusa, Schalthöfer and Selman showed that CPT can define the CFI query if a preorder with colour classes of logarithmic size is available [14]. This is the strongest known positive result concerning the solvability of CFI in CPT.

Our contribution implies that this result cannot be generalised to the CFI problem on unordered input structures: Instances of CFI can be obtained by applying the Cai-Fürer-Immerman construction to any family of connected graphs, in particular also to hypercubes. Since CPT cannot define a sufficiently fine preorder in all hypercubes, and the CFI construction preserves the hypercube-structure, the algorithmic technique from [14] which heavily relies on such preorders cannot be applied to all unordered CFI structures.

Therefore, if CFI on unordered structures is solvable in CPT, entirely new choiceless algorithmic techniques are needed to show this. Otherwise, if CFI is indeed a separating problem for CPT and P, one possible approach to prove this would be to identify further hereditarily finite sets over hypercubes which are not CPT-definable.

Technically, what we show in this paper is a statement concerning the orbit size of certain hereditarily finite objects over hypercubes: For every $n \in \mathbb{N}$, fix a h.f. object representing a preorder in the n-dimensional hypercube. If the colour classes of each preorder are of logarithmic size w.r.t. the hypercube, then the orbit size (w.r.t. the hypercube-automorphisms) of these h.f. objects grows super-polynomially in 2^n, which is the size of the n-dimensional hypercube.

Since CPT is a logic and therefore isomorphism-invariant, it has to define any object together with its entire orbit – if the size of the orbit is not polynomially bounded, then this is not possible in Choiceless Polynomial Time. In fact, we can interpret this non-definability

result as an inherent weakness of choiceless polynomial time computation in general: It holds for any isomorphism-invariant polynomial time (or even polynomial space) computation model on hereditarily finite sets. Hence, should it be the case that CPT fails to capture PTIME because of a super-polynomial orbit argument like this one, we could conclude that the quest for a PTIME-logic should continue with other data structures than hereditarily finite sets.

Finally, we remark that the main combinatorial tool we use in our proof – so-called *supporting partitions* – is taken from [1], where Anderson and Dawar show a correspondence between FPC and Symmetric Circuits. There, it is used for the calculation of orbit sizes of circuit gates. The fact that this tool also helps to understand the symmetries of hereditarily finite objects over hypercubes demonstrates its versatility and usefulness for the study of symmetric objects in general.

2 Choiceless computation and the undefinability of preorders

In this paper, we will not give a definition of CPT, but only state its properties that our lower bound depends on. Thereby, our result also holds for a much broader class of choiceless computation models that includes CPT.
For details on CPT, we refer to the literature: A concise survey on the subject can be found in [8]. It should be noted that there are multiple different ways to formalise CPT: The original definition was via abstract state machines [2], but there are also more "logic-like" presentations such as Polynomial Interpretation Logic (see [9], [16]) and BGS-logic [15]. The latter is essentially a fixed-point logic that allows for the isomorphism-invariant creation and manipulation of *hereditarily finite sets* over the input structure. In fact, it has been shown in [7] that any CPT-program (the words "program" and "sentence" are often used interchangeably in the context of CPT) is equivalent to a sentence in FPC (fixed-point logic with counting) evaluated in the input structure enriched with all the necessary hereditarily finite sets. Therefore, let us make this notion precise.

Hereditarily finite sets and choiceless computation

Let A be a nonempty set. The set of *hereditarily finite objects* over A, $\mathrm{HF}(A)$, is defined as $\bigcup_{i\in\mathbb{N}} \mathrm{HF}_i(A)$, where $\mathrm{HF}_0(A) := A \cup \{\emptyset\}, \mathrm{HF}_{i+1}(A) := \mathrm{HF}_i(A) \cup 2^{\mathrm{HF}_i(A)}$. The size of an h.f. set $x \in \mathrm{HF}(a)$ is measured in terms of its *transitive closure* $\mathrm{tc}(x)$: The set $\mathrm{tc}(x)$ is the least transitive set such that $x \in \mathrm{tc}(x)$. Transitivity means that for every $a \in \mathrm{tc}(x)$, $a \subseteq \mathrm{tc}(x)$.
If the atom set A is the universe of a structure $\mathfrak{A}$, then the action of $\mathrm{Aut}(\mathfrak{A}) \subseteq \mathrm{Sym}(A)$, the automorphism group of $\mathfrak{A}$, extends naturally to $\mathrm{HF}(A)$: For $x \in \mathrm{HF}(A)$, $\pi \in \mathrm{Aut}(\mathfrak{A})$, x^π is obtained from x by replacing each occurrence of an atom a in x with $\pi(a)$.
The *orbit* (w.r.t. the action of $\mathrm{Aut}(\mathfrak{A})$) of an object $x \in \mathrm{HF}(A)$ is the set of all its automorphic images, i.e. $\{x^\pi \mid \pi \in \mathrm{Aut}(\mathfrak{A})\}$. The *stabiliser* $\mathrm{Stab}(x)$ of x is the subgroup $\{\pi \in \mathrm{Aut}(\mathfrak{A}) \mid x^\pi = x\}$.

▶ **Definition 1.** *Let $\mathfrak{A}$ be a finite relational structure with universe A, and $p : \mathbb{N} \longrightarrow \mathbb{N}$ a polynomial. We say that a h.f. object $x \in HF(A)$ is*

- symmetric *(w.r.t. $\mathfrak{A}$) if x is stabilised by all automorphisms of $\mathfrak{A}$, i.e. $Stab(x) = Aut(\mathfrak{A})$;*
- *p-bounded if $|tc(x)| \leq p(|A|)$.*

Every CPT-program comes with an explicit polynomial bound p that limits both the length of its runs as well as the size of the h.f. sets that it may use in the computation. Further, due to its nature as a logic, all operations of CPT are symmetry-invariant. This is

already everything that our lower bound depends on. The following abstract view on the execution of CPT-programs is true regardless of the concrete presentation of CPT, and this level of abstraction is sufficient for the purposes of this paper:

Let Π be a CPT-program with bound p, and $\mathfrak{A}$ be a structure of matching signature. Then the run of Π on $\mathfrak{A}$ is a sequence of h.f. sets $x_1, x_2, ... \in \mathrm{HF}(A)$, each of which is symmetric and p-bounded w.r.t. $\mathfrak{A}$.

Consequently, no CPT-program – and generally, no computation model operating on symmetric p-bounded h.f. sets – can compute a h.f. set x with super-polynomial orbit size because the corresponding stage of the run must contain x along with its entire orbit in order to fulfil the symmetry-condition. Now, we are almost ready to state our general lower bound theorem, which applies to CPT as a special case by the facts just mentioned.

Preorders and colour classes

A *preorder* $\prec$ on a set A induces a partition of A into *colour classes* $C_1, ..., C_m$. A colour class is a set of $\prec$-incomparable elements, and $\prec$ induces a linear order on the colour classes. The canonical representation of such a preorder as a h.f. set is $\{C_1, \{C_2, \{C_3, \{...\}\}\}\}$. However, our lower bound holds for any representation that places elements from the same colour class at the same "nesting depth" within the h.f. set (see Section 4 for the formal definition). This is sufficient because even a representation that does not distinguish colour classes by nesting depth can easily be transformed into the canonical representation above by a CPT-program.

▶ **Theorem 2.** *Let $(H_n)_{n\in\mathbb{N}}$ be the sequence of n-dimensional hypercubes. In each H_n, fix any preorder $\prec_n$ on the vertex set with colour classes of size $\mathcal{O}(n) = \mathcal{O}(\log |H_n|)$. Let x_n be any* symmetric *(w.r.t. H_n) h.f. set over H_n that contains a h.f. representation of $\prec_n$. Then there exists no polynomial p such that x_n is also* p-bounded *w.r.t. the corresponding H_n.*

The proof can be found in Section 6. As already explained, this implies the following nondefinability statement for CPT.

▶ **Corollary 3.** *There is no CPT-program that computes in every hypercube H_n a (h.f. set representation of a) total preorder with colour classes of size $\mathcal{O}(n)$.*

3 Previous work and the significance of undefinable preorders

As already mentioned, our contribution is a non-definability result for a *functional problem*, the computation of certain preorders.
However, our research is motivated by the study of a *decision problem* which is seen as a potential candidate for the separation of CPT from PTIME: The so-called *CFI problem*, that we briefly introduce next. Whether CFI in its general version is solvable in CPT is an open question, but at least for restricted versions, where the structures possess a certain degree of built-in order, it is known to be in CPT. Our non-definability result implies that being able to solve the restricted version of CFI in CPT is of no help for solving CFI in the general case.

The CFI problem

For a detailed account of the CFI problem and the construction of the so-called CFI graphs, we refer the reader to the original paper [3] by Cai, Fürer and Immerman. Here, we only review it to an extent sufficient for our purposes.

Essentially, CFI is the Graph Isomorphism problem on specific pairs of graphs that are obtained by applying the so-called CFI construction to a family of connected graphs, for example, to hypercubes. These are referred to as the *underlying graphs*. The construction

replaces every edge and every vertex of the underlying graph with a gadget. Importantly, the symmetries of the underlying graph are preserved this way. Any underlying graph G can be transformed into an odd and an even CFI graph, G_0 and G_1. It holds $G_0 \ncong G_1$, and there is a simple polynomial time algorithm which can determine, given a CFI graph G_x, whether it is odd or even, i.e. if $G_x \cong G_0$, or $G_x \cong G_1$. This is what the CFI problem asks for.
However, on the logical side, that is, in FO with counting, G_0 and G_1 can only be distinguished with a linear number of variables. As a consequence, no FPC-sentence can solve the CFI-problem (on a suitable class of underlying graphs). Since this very expressive "reference logic" within PTIME fails to solve CFI, this raises the question whether CPT is strong enough to achieve this, or if CFI is indeed a problem that separates CPT from P.

Solving CFI in CPT

If the underlying graphs of the CFI construction satisfy certain properties, then CFI can be solved in CPT:

▶ **Theorem 4** ([14]). *Let $\mathcal{K}$ be the class of connected, preordered graphs $G = (V, E, \prec)$ where the size of each colour class is bounded by $\log |V|$. The CFI problem on underlying graphs in $\mathcal{K}$ can be solved in Choiceless Polynomial Time.*

This is the strongest known positive result concerning CFI and CPT. It is a generalisation of the CPT-algorithm by Dawar, Richerby and Rossman from [7] for the CFI problem on linearly ordered graphs. Note that not the CFI graphs G_0, G_1 are ordered/preordered in these settings, but only the underlying graph G (otherwise, the Immerman-Vardi Theorem could be applied). The order/preorder on G allows for the algorithmic creation of a so-called "super-symmetric" h.f. object with polynomial orbit which makes it possible to determine the parity of the input CFI graph G_x. This object reflects in its structure the preorder on the input, and is therefore not definable in unordered inputs according to our Theorem 2: It can be checked that Theorem 2 not only holds for hypercubes but also for the CFI graphs obtained from them; this is true because $\mathrm{Aut}(G)$ embeds into $\mathrm{Aut}(G_x)$ for any graph G and corresponding CFI graph G_x. Hence, the algorithmic technique that proves Theorem 4 cannot be generalised to the CFI problem on unordered graphs. In fact, any CPT algorithm that is to solve the unordered CFI problem must avoid the construction of a h.f. object whose nesting structure induces a too fine preorder on the input.

We remark that there are of course families of graphs where the undefinability of such preorders is much easier to show than on hypercubes. For instance, on complete graphs, it is clear that the orbit of a preorder with logarithmic colour classes grows super-polynomially. However, the size of any CFI graph G_0 is exponential in the maximal degree of G. Therefore, the polynomial resources of CPT suffice to solve CFI on unordered graphs of linear maximal degree (this is another result from [14]). Hence, complete graphs do not yield hard CFI instances. In contrast, CFI on hypercubes is well-suited as a benchmark for CPT because their degree is logarithmic and thus the CFI construction only increases the size polynomially.

Our lower bound is a first piece of evidence that the CFI problem on hypercubes is hard (and maybe even unsolvable) for CPT and we believe that it deserves further investigation. The results in [7] indirectly suggest a systematic way to do so: Namely, Dawar, Richerby and Rossman showed that – as long as the CFI structures satisfy a certain homogeneity condition – solving the CFI problem in CPT always requires the construction of a h.f. set which contains a large subset of the input structure as atoms. If it were possible to show that no sufficiently large h.f. object over hypercubes has a polynomial orbit, then this could be used to separate CPT from PTIME. Our result is a step in that direction as it suggests that this large object cannot be structurally similar to a preorder.

4 Analysing orbits of hereditarily finite objects over hypercubes

Let $H_n = (V_n, E_n)$ be the n-dimensional hypercube, i.e. $V_n = \{0,1\}^n$, $E_n = \{\{u,v\} \in V_n^2 \mid d(u,v) = 1\}$, where $d(u,v)$ is the Hamming-distance.
It is well-known that its automorphism group $\mathrm{Aut}(H_n)$ is isomorphic to the semidirect product of Sym_n and $(\{0,1\}^n, \oplus)$, where Sym_n is the symmetric group on $[n] = \{1,2,...,n\}$, and $(\{0,1\}^n, \oplus)$ is the group formed by the length-n binary strings together with the bitwise XOR-operation. More precisely, any automorphism $\sigma \in \mathrm{Aut}(H_n)$ corresponds to the pair $(\pi, w) \in \mathrm{Sym}_n \times \{0,1\}^n$ with $\sigma(v) = v^\pi \oplus w$, where $v^\pi = v_{\pi^{-1}(1)} v_{\pi^{-1}(2)} ... v_{\pi^{-1}(n)}$ (i.e. v^π is obtained from v by permuting the positions of the word according to π). This means: $|\mathrm{Aut}(H_n)| = n! \cdot 2^n$. Note that it is the factor $n!$ which makes the size of this group super-polynomial in $|V_n| = 2^n$.

Our main technical theorem, Theorem 13 concerns a fixed sequence of h.f. objects over the n-dimensional hypercubes, $(x_n)_{n\in\mathbb{N}}$, where $x_n \in \mathrm{HF}(V_n)$. We aim for a lower bound on the orbit size of the objects x_n w.r.t. the action of $\mathrm{Aut}(H_n)$ extended to $\mathrm{HF}(V_n)$. For our purposes, it only matters whether this lower bound is super-polynomial in $2^n = |V_n|$, or not. For this question, we can restrict ourselves to automorphisms corresponding to permutation-word pairs of the form $(\pi, 0^n)$, for $\pi \in \mathrm{Sym}_n$. Therefore, for the rest of this paper, we simply let Sym_n act on V_n by permuting the positions of the binary strings as described above. In this sense, Sym_n embeds into $\mathrm{Aut}(H_n)$, and hence, whenever an object $x \in \mathrm{HF}(V_n)$ has a super-polynomial orbit with respect to this action of Sym_n, this is also true with respect to the action of $\mathrm{Aut}(H_n)$.

To sum up, our task is to lower-bound the orbit-sizes of h.f. objects over length-n binary strings with respect to Sym_n acting on the positions of the strings. We do this via the Orbit-Stabiliser Theorem. Let $\rho : \mathrm{Sym}_n \longrightarrow \mathrm{Aut}(H_n)$ denote the aforementioned embedding. For the rest of the paper, let $\mathrm{Stab}_n(x_n)$ and $\mathrm{Orbit}_n(x_n)$ denote the stabiliser and orbit, respectively, of x_n w.r.t. the action of Sym_n on the string positions:

$$\mathrm{Stab}_n(x_n) := \{\pi \in \mathrm{Sym}_n \mid x_n^{\rho(\pi)} = x_n\},\ \mathrm{Orbit}_n(x_n) := \{x_n^{\rho(\pi)} \mid \pi \in \mathrm{Sym}_n\}.$$

(we will usually write x^π instead of $x^{\rho(\pi)}$).

▶ **Proposition 5** (Orbit-Stabiliser).

$$|Orbit_n(x_n)| = \frac{|Sym_n|}{|Stab_n(x_n)|} = \frac{n!}{|Stab_n(x_n)|}.$$

This means that we have to upper-bound $|\mathrm{Stab}_n(x_n)|$. As arbitrary nested sets are not easy to handle, we will not analyse x_n directly, but work with an abstraction that is just a collection of sets over $V_n = \{0,1\}^n$. We call these sets the *levels* of x_n. To define them formally, let $\mathrm{HF}_n := (\mathrm{HF}(V_n), \in)$ be the directed acyclic graph whose nodes are all h.f. objects over V_n and whose edges are given by the element-relation. Then,

$$\mathrm{Level}_i(x_n) := \{v \in V_n \mid \text{in } \mathrm{HF}_n \text{ there is an } \in\text{-path of length } i \text{ from } x_n \text{ to } v\}.$$

We let $I(x_n) \subseteq \mathbb{N}$ be the index-set of the non-empty levels of x_n, i.e. $I(x_n) := \{i \in \mathbb{N} \mid \mathrm{Level}_i(x_n) \neq \emptyset\}$. It is easy to see that any automorphism that stabilises x_n must stabilise each of its levels (not necessarily pointwise, but as a set).

▶ **Proposition 6.**

$$Stab_n(x_n) \subseteq \bigcap_{i \in I(x_n)} Stab_n(Level_i(x_n)).$$

In other words, we have reduced our problem to upper-bounding the size of the simultaneous stabiliser group of a collection of sets of bitstrings. In the next section, we introduce a tool that we need in order to accomplish this: So-called *supporting partitions*.

5 Approximating permutation groups with supporting partitions

The notions and results in this section are mostly taken from the paper on Symmetric Circuits and FPC by Anderson and Dawar [1].

▶ **Definition 7.** *Let $\mathcal{P}$ be a partition of $[n]$.*

- *The* pointwise *stabiliser of $\mathcal{P}$ is $Stab_n^\bullet(\mathcal{P}) := \{\pi \in Sym_n \mid \pi(P) = P \text{ for all } P \in \mathcal{P}\}$.*
- *The* setwise *stabiliser of $\mathcal{P}$ is $Stab_n(\mathcal{P}) := \{\pi \in Sym_n \mid \pi(P) \in \mathcal{P} \text{ for all } P \in \mathcal{P}\}$ (these are all $\pi \in Sym_n$ that induce a permutation on the parts of $\mathcal{P}$).*

▶ **Definition 8** (Supporting Partition, [1])**.** *Let $G \subseteq Sym_n$ be a group. A supporting partition $\mathcal{P}$ of G is a partition of $[n]$ such that $Stab_n^\bullet(\mathcal{P}) \subseteq G$.*

A group $G \subseteq \mathrm{Sym}_n$ may have several supporting partitions but there always exists a unique *coarsest supporting partition*. A partition $\mathcal{P}'$ is as coarse as a partition $\mathcal{P}$, if every part in $\mathcal{P}$ is contained in some part in $\mathcal{P}'$. For any two partitions $\mathcal{P}, \mathcal{P}'$ there exists a finest partition $\mathcal{E}(\mathcal{P}, \mathcal{P}')$ that is as coarse as either of them:

▶ **Definition 9** ([1])**.** *Let $\mathcal{P}, \mathcal{P}'$ be partitions of $[n]$. Let $\sim$ be a binary relation on $[n]$ such that $x \sim y$ iff there exists a part $P \in \mathcal{P}$ or $P \in \mathcal{P}'$ such that $x, y \in P$. Then $\mathcal{E}(\mathcal{P}, \mathcal{P}')$ is the partition of $[n]$ whose parts are the equivalence classes of $[n]$ under the transitive closure of $\sim$.*

As shown in [1], the property of being a supporting partition of a group $G \subseteq \mathrm{Sym}_n$ is preserved under the operation $\mathcal{E}$. Therefore it holds:

▶ **Lemma 10** ([1])**.** *Each permutation group $G \subseteq Sym_n$ has a unique* coarsest supporting partition*, denoted $SP(G)$.*

When we write $\mathrm{SP}(a)$ for $a \in \mathrm{HF}(\{0,1\}^n)$, we mean $\mathrm{SP}(\mathrm{Stab}_n(a))$, that is, the coarsest supporting partition of the stabiliser of a, where – as in the previous section – we consider the stabiliser as the subgroup of Sym_n acting on the positions of the binary strings. Note that if $a \in \{0,1\}^n$, then $\mathrm{SP}(a)$ is just the partition of $[n]$ into $\{k \in [n] \mid a_k = 0\}$ and $\{k \in [n] \mid a_k = 1\}$.

The reason why coarsest supporting partitions are useful for estimating the sizes of certain stabiliser subgroups is the following result:

▶ **Lemma 11** ([1])**.** *Let $G \subseteq Sym_n$ be a group. Then:*

$$Stab_n^\bullet(SP(G)) \subseteq G \subseteq Stab_n(SP(G)).$$

This lemma enables us to upper-bound stabilisers of arbitrary objects in $\mathrm{HF}(\{0,1\}^n)$ by the stabilisers of their supporting partitions.
Finally, because we will frequently need it later in our proof, we define the operation $\sqcap$ as the "intersection" of two partitions:

▶ **Definition 12** (Intersection of partitions)**.** *Let $\mathcal{P}, \mathcal{P}'$ be partitions of $[n]$. The* intersection *$\mathcal{P} \sqcap \mathcal{P}'$ is defined like this:*

$$\mathcal{P} \sqcap \mathcal{P}' := \{\mathcal{P}(k) \cap \mathcal{P}'(k) \mid k \in [n]\}.$$

Here, $\mathcal{P}(k), \mathcal{P}'(k)$ denote the parts of the respective partition that contain k.

6 The Super-Polynomial Orbit Theorem

Our main technical theorem reads as follows:

▶ **Theorem 13.** *Let $(x_n)_{(n\in\mathbb{N})}$ be a sequence with $x_n \in HF(V_n)$ (recall that $V_n = \{0,1\}^n$). Assume that the x_n satisfy the following two properties:*
1. *In each x_n, every $v \in V_n$ occurs as an atom.*
2. *The function $\max_{i\in I(x_n)} |Level_i(x_n)|$ is in $\mathcal{O}(n)$.*

Then, $|Orbit_n(x_n)|$ (as defined in Section 4) grows asymptotically faster than any polynomial in $2^n = |V_n|$.

From this, Theorem 2 follows because – as discussed in Section 2 – the canonical h.f. set representation of a preorder with logarithmic colour classes satisfies the two conditions of Theorem 13, and because any *symmetric* (see Definition 1) h.f. object that contains x_n must necessarily contain $\mathrm{Orbit}_n(x_n)$, too.
We start to explain the proof idea of Theorem 13 by stating the following summary of Proposition 6 and Lemma 11:

▶ **Corollary 14.**

$$Stab_n(x_n) \subseteq \bigcap_{i\in I(x_n)} Stab_n(Level_i(x_n))) \subseteq \bigcap_{i\in I(x_n)} Stab_n(SP(Level_i(x_n))).$$

We are going to employ the Orbit-Stabiliser Theorem in order to obtain our lower bound for the orbit size. Hence, we need to bound $|\mathrm{Stab}_n(x_n)|$ from above, and Corollary 14 already indicates the basic principle of our proof: Splitting up x_n into its levels and analysing the stabilisers of their respective supporting partitions.
Our analysis of $|\mathrm{Stab}_n(x_n)|$ is divided into two main cases that we treat separately. The distinction is with respect to the maximum size of the coarsest support of any level of x_n, viewed as a function of n:

Let $B_n \subseteq \{0,1\}^n$ be the level of x_n such that $|\mathrm{SP}(B_n)|$ (i.e. its number of parts) is maximal in $\{|\mathrm{SP}(\mathrm{Level}_i(x_n))| \mid i \in I(x_n)\}$. Then the two cases we distinguish are:
(1) The maximal level-support size grows sublinearly: $|\mathrm{SP}(B_n)| \in o(n)$.
(2) The maximal level-support size grows linearly: $|\mathrm{SP}(B_n)| \in \Theta(n)$.

We deal with the two cases in the next two subsections. Their results are summarised in Lemma 15 and Lemma 21. Together they imply the theorem. Due to space restrictions, we can only give proof sketches for most lemmas; for some of them, full proofs can be found in the appendix. In the following lemmas, we always refer to the objects and the setting of Theorem 13, as well as to the set level B_n just defined.

6.1 The case of sublinearly bounded supports

The result of this subsection is:

▶ **Lemma 15.** *Assume the following three conditions hold:*
1. *In each x_n, every $v \in V_n$ occurs as an atom.*
2. *The function $\max_{i\in I(x_n)} |Level_i(x_n)|$ is in $\mathcal{O}(n)$.*
3. *$|SP(B_n)| \in o(n)$.*

Then the orbit size of x_n w.r.t. Sym_n acting on the positions of the binary strings grows faster than any polynomial in 2^n.

We prove this lemma on the next few pages. From now on, we use the abbreviation $\mathrm{SP}_i(x_n) := \mathrm{SP}(\mathrm{Level}_i(x_n))$. Let us begin by outlining the proof idea. We have to bound $\mathrm{Stab}_n(x_n) \subseteq \bigcap_{i\in I(x_n)} \mathrm{Stab}_n(\mathrm{SP}_i(x_n))$ (see Corollary 14). Hence, we have to count the permutations in Sym_n that simultaneously stabilise the supports of the levels of x_n.
For a level $i \in I(x_n)$, $\mathrm{Sym}(\mathrm{SP}_i(x_n))$ denotes the symmetric group on the parts of $\mathrm{SP}_i(x_n)$ (in contrast, Sym_n is the symmetric group on the set $[n]$ that underlies this partition). Every $\pi \in \mathrm{Sym}_n$ that stabilises $\mathrm{SP}_i(x_n)$ as a set *induces* (or *realises*) a $\sigma \in \mathrm{Sym}(\mathrm{SP}_i(x_n))$ in the sense that $\sigma(P) = \{\pi(k) \mid k \in P\} \in \mathrm{SP}_i(x_n)$ for all $P \in \mathrm{SP}_i(x_n)$. This can also be extended to a set $J \subseteq I(x_n)$ of several levels: Every $\pi \in \bigcap_{i\in J} \mathrm{Stab}_n(\mathrm{SP}_i(x_n))$ induces a $\overline{\sigma} \in \bigtimes_{i\in J} \mathrm{Sym}(\mathrm{SP}_i(x_n))$. Here, $\overline{\sigma}$ is the tuple of permutations that π realises simultaneously on the parts of the respective $\mathrm{SP}_i(x_n)$.

Now in order to bound $|\mathrm{Stab}_n(x_n)|$, we will choose a subset $J \subseteq I(x_n)$ with certain properties that will enable us to bound two quantities: Firstly, each $\overline{\sigma} \in \bigtimes_{i\in J} \mathrm{Sym}(\mathrm{SP}_i(x_n))$ that can be realised by a $\pi \in \mathrm{Stab}_n(x_n)$ will only have a small number of distinct such realisations. Secondly, there will be a bound on the number of such $\overline{\sigma}$ that can be realised by a $\pi \in \mathrm{Stab}_n(x_n)$ at all. The product of these two bounds is then an upper bound for $|\mathrm{Stab}_n(x_n)|$.

We begin with a lemma that generally relates the number of possible distinct realisations of a given $\overline{\sigma} \in \bigtimes_{i\in[m]} \mathrm{Sym}(\mathrm{SP}(A_i))$, for sets $A_1, ..., A_m \subseteq \{0,1\}^n$, with the partition $\bigsqcap_{i=1}^m \mathrm{SP}(A_i)$ (recall Definition 12 for the meaning of $\sqcap$).

▶ **Lemma 16.** *Let $A_1, ..., A_m \subseteq \{0,1\}^n$ be a collection of sets of bitstrings. Fix any simultaneous permutation $\overline{\sigma}$ of the parts of the supports of the sets, i.e. $\overline{\sigma} \in \bigtimes_{i=1}^m Sym(SP(A_i))$. There exists a $\vartheta_{\overline{\sigma}} \in Sym(\bigsqcap_{i=1}^m SP(A_i))$ such that every $\pi \in Sym_n$ that realises $\overline{\sigma}$ also realises $\vartheta_{\overline{\sigma}}$.*

Proof sketch. Via induction on m. If $m = 1$, then the desired $\vartheta_{\overline{\sigma}}$ is just $\overline{\sigma} \in \mathrm{Sym}(\mathrm{SP}(A_1))$. For the induction step, let there be $m+1$ sets $A_1, ..., A_{m+1}$, and let $\overline{\sigma} \in \bigtimes_{i=1}^{m+1} \mathrm{Sym}(\mathrm{SP}(A_i))$ be fixed. Since every $\pi \in \mathrm{Sym}_n$ that realises $\overline{\sigma}$ in particular realises the first m entries in $\overline{\sigma}$, the induction hypothesis gives us a fixed permutation in $\mathrm{Sym}(\bigsqcap_{i=1}^m \mathrm{SP}(A_i))$ that each such π has to realise. Further, π has to realise $\overline{\sigma}_{m+1} \in \mathrm{Sym}(\mathrm{SP}(A_{m+1}))$. Putting these constraints on π together, the desired $\vartheta_{\overline{\sigma}} \in \mathrm{Sym}(\bigsqcap_{i=1}^{m+1} \mathrm{SP}(A_i))$ is obtained. ◀

So, intuitively speaking, the finer the partition $\bigsqcap_{i=1}^m \mathrm{SP}(A_i)$ is, the fewer realisations exist for any $\overline{\sigma} \in \bigtimes_{i=1}^m \mathrm{Sym}(\mathrm{SP}(A_i))$. Therefore, we will aim to select a subset of the levels of x_n such that the intersection over the supports is as fine as possible. More precisely, we would like it to consist of many singleton parts. For the rest of this subsection we denote by $S_n \subseteq [n]$ the set of positions which are in singleton parts in $\bigsqcap_{i\in I(x_n)} \mathrm{SP}_i(x_n)$, i.e.

$$S_n := \{k \in [n] \mid \{k\} \in \bigsqcap_{i\in I(x_n)} \mathrm{SP}_i(x_n)\}.$$

It turns out that there can only be few positions which are *not* in singleton parts in $\bigsqcap_{i\in I(x_n)} \mathrm{SP}_i(x_n)$; this is a consequence of the assumption that x_n contains every element of $\{0,1\}^n$, together with the size bound on the levels:

▶ **Lemma 17.** *Assume that $|Level_i(x_n)| \in \mathcal{O}(n)$ for each level i of x_n. Further, assume that in each x_n, every element of $\{0,1\}^n$ occurs as an atom. Then, for large enough n:*

$$|[n] \setminus S_n| < 8 \log n.$$

Proof sketch. Assume: $|[n] \setminus S_n| \geq 8 \log n$. We show that this entails the existence of a level $A \subseteq \{0,1\}^n$ in the object x_n such that $|A|$ is greater than $\mathcal{O}(n)$, which is a contradiction. The partition $\mathrm{SP}(A)$ is as least as coarse as $\bigsqcap_{i \in I(x_n)} \mathrm{SP}_i(x_n)$ and has therefore a certain number of positions within non-singleton parts according to our assumption. Permuting the positions within the non-singleton parts of $\mathrm{SP}(A)$ leaves the set A intact, by definition of supporting partitions. Hence, if A contains a string a with a balanced number of zeroes and ones within each $P \in \mathrm{SP}(A)$ with $|P| \geq 2$, it can be calculated that a has more than $\mathcal{O}(n)$ images under the mentioned permutations. Because every $a \in \{0,1\}^n$ occurs somewhere in x_n, such a level A of x_n indeed exists. ◀

We proceed to construct the announced subset $J \subseteq I(x_n)$ of the levels of x_n. Its two properties that are stated in the next lemma are crucial to bound $|\bigcap_{i \in J} \mathrm{Stab}_n(\mathrm{SP}_i(x_n))|$. A full proof of the lemma is included in the appendix.

▶ **Lemma 18.** *Let $f(n) \in o(n)$ such that for all levels $i \in I(x_n)$, $|SP_i(x_n)| \leq f(n)$. There exists a subset $J \subseteq I(x_n)$ of the levels of x_n with the following two properties:*

(1) *Every position in S_n is also in a singleton part of $\bigsqcap_{j \in J} SP_j(x_n)$.*

(2) *The following bound for the number of realisable simultaneous permutations of the supporting partitions holds:*

$$\left|\{\overline{\sigma} \in \bigtimes_{j \in J} Sym(SP_j(x_n)) \mid \text{there is a } \pi \in Sym_n \text{ that realises } \overline{\sigma}\}\right| \leq (f(n)!)^{n/(f(n)-1)} \cdot 2^n$$

Proof sketch. Construct the set J stepwise, starting with $J^0 = \emptyset$, and adding a new level j_i in each step i, such that $\bigsqcap_{j \in J^i} \mathrm{SP}_j(x_n)$ is a strict refinement of $\bigsqcap_{j \in J^{i-1}} \mathrm{SP}_j(x_n)$. This is done until property (1) is satisfied. Let

$$k_i := \left|\bigsqcap_{j \in J^i} \mathrm{SP}_j(x_n)\right| - \left|\bigsqcap_{j \in J^{i-1}} \mathrm{SP}_j(x_n)\right|.$$

The main part of the proof is to show that in step $i+1$, there are at most $(k_{i+1}+1)!$ permutations in $\mathrm{Sym}(\mathrm{SP}_{j_{i+1}}(x_n))$ that can be realised by some $\pi \in \mathrm{Sym}_n$ simultaneously with any other given $\overline{\sigma} \in \bigtimes_{j \in J^i} \mathrm{Sym}(\mathrm{SP}_j(x_n))$. Once this is established, we know that the set of simultaneous permutations from property (2) has size at most $\prod_{i=1}^{s}(k_i+1)!$. Further, for all i we have $|\mathrm{Sym}(\mathrm{SP}_i(x_n))| \leq f(n)!$. Using the fact that $\sum_{i=1}^{s} k_i \leq n$, one can now make a typical "redistribute weight argument" to show that the mentioned product of factorials is maximised if each k_i is either 1 or $f(n)-1$. This leads to the bound stated in property (2). ◀

▶ **Corollary 19.** *Assume the following three conditions hold:*

1. *In each x_n, every $v \in V_n$ occurs as an atom.*
2. *The function $\max_{i \in I(x_n)} |Level_i(x_n)|$ is in $\mathcal{O}(n)$.*
3. *$|SP(B_n)| \in o(n)$.*

Then, for sufficiently large n:

$$|Stab_n(x_n)| \leq (f(n)!)^{n/(f(n)-1)} \cdot 2^n \cdot (8 \log n)!$$

Proof. Consider the set $J \subseteq I(x_n)$ that exists by Lemma 18. By Corollary 14, every $\pi \in \mathrm{Stab}_n(x_n)$ induces a tuple of permutations $\overline{\sigma} \in \bigtimes_{i \in I(x_n)} \mathrm{Sym}(\mathrm{SP}_i(x_n))$, so in particular it also induces a $\overline{\sigma} \in \bigtimes_{i \in J} \mathrm{Sym}(\mathrm{SP}_i(x_n))$. By Lemma 18, there are at most $(f(n)!)^{n/(f(n)-1)} \cdot 2^n$ possibilities for such a $\overline{\sigma}$. Furthermore, each such $\overline{\sigma}$ can be realised by at most $(8 \log n)!$

distinct permutations $\pi \in \mathrm{Stab}_n(x_n)$: Due to Lemma 16 and property (1) of J (see Lemma 18), every π realising $\overline{\sigma}$ permutes the positions in S_n in the same way, and according to Lemma 17, there remain at most $8 \log n$ positions which may be permuted arbitrarily by π (that is, if all positions in $[n] \setminus S_n$ form a single part in $\bigsqcap_{i \in J} \mathrm{SP}_i(x_n)$). ◀

With this, we can estimate the asymptotic behaviour of $|\mathrm{Orbit}_n(x_n)|$, which proves Lemma 15.

▶ **Lemma 20.** *Under the assumptions of Corollary 19, $|Orbit_n(x_n)|$ can be estimated as follows: For any $k \in \mathbb{N}$, the limit*

$$\lim_{n\to\infty} \frac{|Orbit_n(x_n)|}{2^{kn}} = \lim_{n\to\infty} \frac{n!}{|Stab_n(x_n)| \cdot 2^{kn}} \geq \lim_{n\to\infty} \frac{n!}{(f(n)!)^{n/(f(n)-1)} \cdot 2^n \cdot (8 \log n)! \cdot 2^{kn}}$$

does not exist. That is to say, the orbit of x_n w.r.t. the action of Sym_n grows super-polynomially in 2^n.

Proof sketch. Replace all factorials with the Stirling Formula $n! \approx \sqrt{2\pi n} \cdot \left(\frac{n}{e}\right)^n$. With this, compute a lower bound for the above fraction that can be seen to tend to infinity as n grows. ◀

6.2 The case of linearly-sized supports

This subsection is dedicated to proving the following result for the case that $|\mathrm{SP}(B_n)| \in \Theta(n)$. In this case, we only need to analyse the orbit size of the level B_n of x_n with the largest supporting partition (see the beginning of Section 6 again for the definition of B_n).

▶ **Lemma 21.** *Assume that the following conditions hold for B_n:*

1. $|B_n| \in \mathcal{O}(n)$.
2. $|SP(B_n)| \in \Theta(n)$.

Then the orbit size of B_n (and therefore also of x_n) w.r.t. the action of Sym_n on the positions of the binary strings grows faster than any polynomial in 2^n.

Proving this lemma requires a case distinction again. The relevant measure here is the number of *singleton* parts in $\mathrm{SP}(B_n)$. Firstly, we show that if the number of singleton parts in $\mathrm{SP}(B_n)$ grows sublinearly in n, while the total number of parts $|\mathrm{SP}(B_n)|$ is linear, the stabiliser of $\mathrm{SP}(B_n)$ is small enough. This can be seen solely from the properties of the partition $\mathrm{SP}(B_n)$.

The difficult part of the proof is the case where the number of singleton parts grows linearly. In the worst case, $\mathrm{SP}(B_n)$ consists only of singletons; then, $\mathrm{Stab}_n(\mathrm{SP}(B_n)) = \mathrm{Sym}_n$. We solve this by not only looking at the partition $\mathrm{SP}(B_n)$ itself but also at properties of the set B_n that can be inferred from its supporting partition.
In the following, we always denote by $S_n \subseteq [n]$ the set of positions that are in singleton parts of $\mathrm{SP}(B_n)$, i.e.

$$S_n := \{k \in [n] \mid \{k\} \in \mathrm{SP}(B_n)\}.$$

(note that the definition of S_n was slightly different in the last subsection).

Subcase 1: Sublinear number of singleton parts

Let us begin with the easier case, where the number of singleton parts in $\mathrm{SP}(B_n)$ grows sublinearly. The size of $\mathrm{Stab}_n(\mathrm{SP}(B_n))$ can generally be bounded as follows:

▶ **Lemma 22.** *Let $s_n := |S_n|$, and $t_n := |SP(B_n)| - s_n$.*

$$|Stab_n(SP(B_n))| \leq s_n! \cdot t_n! \cdot (n - 2(t_n - 1))! \cdot 2^{t_n}.$$

Proof. The factors $s_n!$ and $t_n!$ account for the possible permutations of the parts: All the singleton parts of $\mathrm{SP}(B_n)$ can be mapped to each other, and every non-singleton part can at most be mapped to every other non-singleton part. An upper bound on the number of permutations within the non-singleton parts is $\ell_1! \cdot \ell_2! \cdot \ldots \cdot \ell_{t_n}!$, where the ℓ_i are the sizes of these parts. This product of factorials is maximised if one value ℓ_p is as large as possible ($\leq n - 2(t_n - 1)$), and $\ell_i = 2$ for all $i \neq p$. This is a standard "redistribute weight argument", which is also used in [1] multiple times. ◀

▶ **Corollary 23.** *Let $f(n) \in o(n)$ be a function such that $s_n \leq f(n)$ and let $c \leq 1$ be a positive constant such that $|SP(B_n)| \geq c \cdot n$ for large enough n. Then, for large enough n, the following bound holds:*

$$|Stab_n(SP(B_n))| \leq f(n)! \cdot 2^n \cdot \max\Big\{(n/2)! \cdot 2, (cn - f(n))! \cdot ((1 - 2c)n + 2f(n) + 2)!\Big\}.$$

Proof. We plug in the right values for s_n and $t_n = |\mathrm{SP}(B_n)| - s_n$ into Lemma 22, and use the simple bound $2^{t_n} \leq 2^n$. We have $s_n \leq f(n)$ by assumption. As $|\mathrm{SP}(B_n)| \geq c \cdot n$, and because every non-singleton part has at least two elements, we can bound t_n as follows:

$$c \cdot n - f(n) \leq t_n \leq \frac{n}{2}.$$

The bound from Lemma 22 contains a product of two factorials which both depend on t_n. By a redistribute-weight argument, one can see that this product is maximised if the two factorials are maximally imbalanced. This happens if t_n attains its maximum or minimum. ◀

This directly leads to a super-polynomial orbit: In the next lemma, we calculate the growth of $|\mathrm{Orbit}_n(x_n)| \geq \frac{n!}{|\mathrm{Stab}_n((\mathrm{SP}(B_n))|}$ (this is due to Lemma 11), using the stabiliser-bound from Corollary 23.

▶ **Lemma 24.** *Let $f(n) \in o(n)$ be a function such that $s_n \leq f(n)$ and $c \leq 1$ be a positive constant such that $|SP(B_n)| \geq c \cdot n$ for large enough n. Then for any $k \in \mathbb{N}$, the limit*

$$\lim_{n\to\infty} \frac{n!}{f(n)! \cdot 2^n \cdot \max\Big\{(n/2)! \cdot 2, (cn - f(n))! \cdot ((1 - 2c)n + 2f(n) + 2)!\Big\} \cdot 2^{kn}}$$

does not exist. That is to say, the orbit of B_n w.r.t. the action of Sym_n grows super-polynomially in 2^n.

Proof sketch. Similar to Lemma 20. ◀

This proves Lemma 21 under the assumption that $|S_n| \in o(n)$.

Subcase 2: Linear number of singleton parts

The idea for this case is similar to how we solved the case of sublinear supports. There, we related simultaneous permutations of the parts of the supports $\mathrm{SP}_i(x_n)$ to their realisations in Sym_n. Now we do the same with respect to permutations of the elements of B_n: Let $\mathrm{Sym}(B_n)$ be the group of all permutations of the strings in B_n. For $\pi \in \mathrm{Sym}_n$ and $\sigma \in \mathrm{Sym}(B_n)$, we say that π *realises* or *induces* σ, if $b^\pi = \sigma(b)$ for every $b \in B_n$. The aim is to show that only a bounded number of $\sigma \in \mathrm{Sym}(B_n)$ can be realised by a permutation $\pi \in \mathrm{Stab}_n(B_n)$ at all, and that each such σ only has a small number of realisations. In total, this yields a bound on $|\mathrm{Stab}_n(B_n)|$.

We will construct a subset $A \subseteq B_n$ such that: Any $\sigma \in \mathrm{Sym}(B_n)$ whose preimages are fixed on A can only be realised by few $\pi \in \mathrm{Sym}_n$, and A is small compared to B_n. This ensures that there are not too many ways to specify a $\sigma \in \mathrm{Sym}(B_n)$ on A (if $|B_n| \in \mathcal{O}(n)$, there are $\leq n^{|A|}$ options to fix $\sigma^{-1}(a)$ for all $a \in A$).
First of all, we show how to bound the number of possible realisations of any $\sigma \in \mathrm{Sym}(B_n)$ if σ is fixed on some subset $A \subseteq B_n$. The next lemma is similar to Lemma 16. We omit the proof as it is quite analogous.

▶ **Lemma 25.** *Let $B \subseteq \{0,1\}^n, A \subseteq B$. Let an injective mapping $p : A \longrightarrow B$ be given. Write $\bigsqcap A := \bigsqcap_{a \in A} SP(a)$.*
There is an assignment of positions to parts $Q_p : [n] \longrightarrow \bigsqcap A$ with the property that $|Q_p^{-1}(P)| = |P|$ for every $P \in \bigsqcap A$, and such that:
Every $\pi \in Sym_n$ realising any $\sigma \in Sym(B)$ with $\sigma^{-1}(a) = p(a)$ (if such a π exists) satisfies: $\pi(k) \in Q_p(k)$ for all $k \in [n]$.

We will mainly need this lemma for the restriction of the parts in $\bigsqcap A$ to the positions in S_n. Therefore, we state the following important corollary:

▶ **Corollary 26.** *Let $A \subseteq B_n$ be arbitrary, and let an injective mapping $p : A \longrightarrow B_n$ be given. Then every $\pi \in Sym_n$ that realises a $\sigma \in Sym(B_n)$ with $\sigma^{-1}(a) = p(a)$ for all $a \in A$ satisfies:*

$$\pi^{-1}(P \cap S_n) = Q_p^{-1}(P) \cap S_n \text{ for all } P \in \bigsqcap A,$$

where $Q_p : [n] \longrightarrow \bigsqcap A$ is the assignment that exists by the preceding lemma.

Proof. Lemma 25 says that $\pi^{-1}(P) = Q_p^{-1}(P)$. Since π realises a permutation in $\mathrm{Sym}(B_n)$, $\pi \in \mathrm{Stab}_n(B_n)$. Hence, by Lemma 11, $\pi \in \mathrm{Stab}_n(\mathrm{SP}(B_n))$. This means that $\pi(S_n) = S_n$, as singleton parts can only be mapped to singleton parts. Consequently, it must be the case that $\pi^{-1}(P \cap S_n) = Q_p^{-1}(P) \cap S_n$. ◀

Next, we select our desired subset $A \subseteq B_n$. It will be such that the partition $\bigsqcap A = \bigsqcap_{a \in A} \mathrm{SP}(a)$ is quite fine on S_n. In order to guarantee that A is much smaller than B_n, we only require a relaxed, but more complicated, notion of "fineness" here. The proof of the next lemma can be found in the appendix.

▶ **Lemma 27.** *There exists a subset $A \subseteq B_n$ of size $|A| \leq \frac{|S_n|}{2}$ such that for each part $P \in \bigsqcap A$, one of the following two statements is true:*

1. *$|P \cap S_n| \leq 2$; **or:***
2. *$|P \cap S_n| > 2$ and for every $b \in B_n \setminus A$, one of these two conditions holds:*
   - *b is constant on $P \cap S_n$; **or***
   - *$b[P \cap S_n]$ is* imbalanced *and, for every $P' \in \bigsqcap A$ with $P' \neq P$, $|P' \cap S_n| > 2$, b is constant on $P' \cap S_n$.*

By $b[P \cap S_n]$ we mean the substring of b at the positions in $P \cap S_n$, and being imbalanced *means that $b[P \cap S_n]$ contains exactly one* 0 *and there is a* 1 *at all other positions, or vice versa (exactly one* 1 *and the rest* 0*).*

Proof sketch. Construct A stepwise, starting with $A^0 = \emptyset$ and adding a new string a_i in each step i. Choose a_{i+1} such that progress is made. That means, a_{i+1} should split some part ($P \cap S_n$), for a $P \in \bigsqcap A^i$ with $|P \cap S_n| > 2$ ("split" means, a_{i+1} is non-constant on $P \cap S_n$). However, we take care that a_{i+1} either splits two parts, or if it splits only one part, it does not split off a singleton part. This ensures that at most $|S_n|/2$ such construction steps can be performed. If no such $a_{i+1} \in B_n \setminus A^i$ exists, then the constructed set fulfils the properties stated in the lemma. ◀

Before we can use this to bound $|\mathrm{Stab}_n(B_n)|$, we need one more lemma concerning those parts $P \in \bigsqcap A$ with $|P \cap S_n| > 2$. We show that any $\pi \in \mathrm{Stab}_n(B_n)$ is already fully determined when it is specified only on the parts $P \in \bigsqcap A$ with $|P \cap S_n| \leq 2$. The full proof of this is also in the appendix.

▶ **Lemma 28.** *Let $A \subseteq B_n$ be the subset that exists by Lemma 27, and let $p : A \longrightarrow B_n$ be an injective function. Let*

$$\Gamma_p := \{\pi \in \mathit{Stab}_n(B_n) \mid p(a)^\pi = a \text{ for all } a \in A\}.$$

Further, let

$$P_{>2} := \{k \in S_n \mid |P(k) \cap S_n| > 2, \text{ where } P(k) \in \bigsqcap A \text{ is the part that } k \text{ is in}\}.$$

Then for any $\pi, \pi' \in \Gamma_p$ such that $\pi^{-1}|_{([n]\setminus P_{>2})} = \pi'^{-1}|_{([n]\setminus P_{>2})}$, it also holds $\pi^{-1}|_{P_{>2}} = \pi'^{-1}|_{P_{>2}}$.

Proof sketch. Assume for a contradiction the existence of $\pi, \pi' \in \Gamma_p$ such that their preimages are the same on $[n] \setminus P_{>2}$ but there is a position x such that $\pi(x) \in P_{>2}$ and $\pi(x) \neq \pi'(x)$. It can then be shown – using the second statement of Lemma 27 and the fact that $\pi, \pi' \in \mathrm{Stab}_n(B_n)$ – that the transposition $(\pi(x)\ \pi'(x))$ is also an element of $\mathrm{Stab}_n(B_n)$. This, however, is a contradiction to the fact that $\pi(x), \pi'(x)$ are in distinct singleton parts in $\mathrm{SP}(B_n)$, which is the coarsest possible supporting partition. ◀

▶ **Lemma 29.** *Let c be a constant such that $|B_n| \leq c \cdot n$ (for large enough n). Then, for large enough n, it holds:*

$$|\mathit{Stab}_n(B_n)| \leq (2cn)^{|S_n|/2} \cdot (n - |S_n|)!$$

Proof. Let $A \subseteq B_n$ be the subset of B_n whose existence is stated in Lemma 27. Fix any injective function $p : A \longrightarrow B_n$. Let Γ_p and $P_{>2}$ be as in Lemma 28.
We bound $|\Gamma_p|$ by counting the number of possible $\pi \in \Gamma_p$. We know by Lemma 28 that we only have to count the number of possibilities to choose the preimages of the elements in $[n] \setminus P_{>2}$. For every part $P \in \bigsqcap A$ with $|P \cap S_n| \leq 2$, we know by Corollary 26 that $\pi^{-1}(P \cap S_n) \subseteq S_n$ is the same fixed set of size ≤ 2 for all $\pi \in \Gamma_p$, so we only have two options how π^{-1} can behave on $P \cap S_n$. The number of such parts P is at most $|S_n|/2$.
For $i \in [n] \setminus S_n$, we can only say that $\pi^{-1}(i) \notin S_n$ (by Lemma 11). Hence, every $\pi \in \Gamma_p$ can in principle permute the set $[n] \setminus S_n$ arbitrarily. In total, we conclude:

$$|\Gamma_p| \leq 2^{(|S_n|/2)} \cdot (n - |S_n|)!$$

This is for a fixed function p. The number of possible choices for p is bounded by $(cn)^{|S_n|/2}$, since $|A| \leq |S_n|/2$ (Lemma 27) and we are assuming $|B_n| \leq cn$.
Every $\pi \in \mathrm{Stab}_n(B_n)$ must occur in at least one of the sets Γ_p for some choice of p, so indeed, $(2cn)^{|S_n|/2} \cdot (n - |S_n|)!$ is an upper bound for $|\mathrm{Stab}_n(B_n)|$. ◀

Based on Lemma 29, the orbit size of B_n can be estimated:

▶ **Lemma 30.** *Let c be a constant such that $|B_n| \leq c \cdot n$, and $\delta > 0$ be a constant such that $|S_n| \geq \delta \cdot n$ (for large enough n). Then for any $k \in \mathbb{N}$, the limit*

$$\lim_{n\to\infty} \frac{|Orbit_n(B_n)|}{2^{kn}} = \lim_{n\to\infty} \frac{n!}{|Stab_n(B_n)| \cdot 2^{kn}} \geq \lim_{n\to\infty} \frac{n!}{(2cn)^{(\delta n)/2} \cdot ((1-\delta)n)! \cdot 2^{kn}}$$

does not exist. That is to say, the orbit of B_n w.r.t. the action of Sym_n grows super-polynomially in 2^n.

Proof sketch. Again a calculation using Stirling's approximation for the factorials. ◀

This lemma together with Lemma 24 proves Lemma 21.

7 Concluding remarks and future research

A question that remains open is what exactly is the threshold of "fineness" of a preorder where the orbit size changes from super-polynomial to polynomial. In other words: What is the largest colour class size for which our Super-Polynomial Orbit Theorem for hypercubes still holds?
One can check that all parts of our proof can be modified such that it also goes through if we allow colour classes (i.e. levels) of size $o(n^2)$. If the size is in $\Theta(n^2)$, though, the bound in Lemma 29 becomes too large for Lemma 30 to hold.
On the other hand, the finest preorder with a polynomial orbit that we know so far is one where the colour class sizes are in $\mathcal{O}(2^n/\sqrt{n})$: It corresponds to the partition of $\{0,1\}^n$ according to Hamming-weight. Obviously, this is precisely the orbit-partition of the vertex-set (w.r.t. the action of Sym_n on the positions). Its largest colour class has size $\binom{n}{n/2} \in \Theta(2^n/\sqrt{n})$.
Determining the finest preorder that is in principle CPT-definable in hypercubes would potentially allow to better judge whether a preorder-based CPT-algorithm like the one in [14] can at all be a candidate for a solution of the unordered CFI problem.
Moreover, it would be helpful to identify further h.f. objects that are undefinable in hypercubes for symmetry reasons.

References

1 Matthew Anderson and Anuj Dawar. On symmetric circuits and fixed-point logics. *Theory of Computing Systems*, 60(3):521–551, 2017. doi:10.1007/s00224-016-9692-2.
2 Andreas Blass, Yuri Gurevich, and Saharon Shelah. Choiceless polynomial time. *Annals of Pure and Applied Logic*, 100(1-3):141–187, 1999. doi:10.1016/S0168-0072(99)00005-6.
3 Jin-yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12:389–410, 1992. doi:10.1007/BF01305232.
4 Ashok K Chandra and David Harel. Structure and complexity of relational queries. In *21st Annual Symposium on Foundations of Computer Science (sfcs 1980)*, pages 333–347. IEEE, 1980. doi:10.1109/SFCS.1980.41.
5 Anuj Dawar. The nature and power of fixed-point logic with counting. *ACM SIGLOG News*, 2(1):8–21, 2015.

6 Anuj Dawar, Martin Grohe, Bjarki Holm, and Bastian Laubner. Logics with rank operators. In *2009 24th Annual IEEE Symposium on Logic In Computer Science*, pages 113–122. IEEE, 2009. doi:10.1109/LICS.2009.24.

7 Anuj Dawar, David Richerby, and Benjamin Rossman. Choiceless polynomial time, counting and the Cai–Fürer–Immerman graphs. *Annals of Pure and Applied Logic*, 152(1-3):31–50, 2008. doi:10.1016/j.apal.2007.11.011.

8 Erich Grädel and Martin Grohe. Is Polynomial Time Choiceless? In *Fields of Logic and Computation II*, pages 193–209. Springer, 2015. doi:10.1007/978-3-319-23534-9_11.

9 Erich Grädel, Wied Pakusa, Svenja Schalthöfer, and Łukasz Kaiser. Characterising Choiceless Polynomial Time with First-order Interpretations. In *Proceedings of the 30th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 677–688, 2015. doi:10.1109/LICS.2015.68.

10 Martin Grohe. The quest for a logic capturing PTIME. In *2008 23rd Annual IEEE Symposium on Logic in Computer Science*, pages 267–271. IEEE, 2008. doi:10.1109/LICS.2008.11.

11 Yuri Gurevich. Logic and the Challenge of Computer Science. In *Current Trends in Theoretical Computer Science*. Computer Science Press, 1988.

12 Neil Immerman. Relational queries computable in polynomial time. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 147–152, 1982. doi:10.1145/800070.802187.

13 Wied Pakusa. *Linear Equation Systems and the Search for a Logical Characterisation of Polynomial Time*. PhD thesis, RWTH Aachen, 2015.

14 Wied Pakusa, Svenja Schalthöfer, and Erkal Selman. Definability of Cai-Fürer-Immerman problems in Choiceless Polynomial Time. *ACM Transactions on Computational Logic (TOCL)*, 19(2):1–27, 2018. doi:10.1145/3154456.

15 Benjamin Rossman. Choiceless computation and symmetry. In *Fields of Logic and Computation, Essays Dedicated to Yuri Gurevich on the Occasion of His 70th Birthday*, volume 6300 of *Lecture Notes in Computer Science*, pages 565–580. Springer, 2010. doi:10.1007/978-3-642-15025-8_28.

16 Svenja Schalthöfer. *Choiceless Computation and Logic*. PhD thesis, RWTH Aachen, 2020.

17 Moshe Y Vardi. The complexity of relational query languages. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 137–146, 1982. doi:10.1145/800070.802186.

8 Appendix

8.1 Proof of Lemma 18

▶ **Lemma 18.** *Let $f(n) \in o(n)$ such that for all levels $i \in I(x_n)$, $|SP_i(x_n)| \leq f(n)$. There exists a subset $J \subseteq I(x_n)$ of the levels of x_n with the following two properties:*

(1) *Every position in S_n is also in a singleton part of $\prod_{j\in J} SP_j(x_n)$.*

(2) *The following bound for the number of realisable simultaneous permutations of the supporting partitions holds:*

$$\left|\{\overline{\sigma} \in \bigtimes_{j\in J} Sym(SP_j(x_n)) \mid \text{there is a } \pi \in Sym_n \text{ that realises } \overline{\sigma}\}\right| \leq (f(n)!)^{n/(f(n)-1)} \cdot 2^n$$

Proof. We construct J stepwise, starting with $J^0 := \emptyset$ and adding one new level $j_i \in I(x_n)$ in each step $i \geq 1$ in such a way that

$$\left|\bigsqcap_{j\in J^{i-1}} \mathrm{SP}_j(x_n) \sqcap \mathrm{SP}_{j_i}(x_n)\right| > \left|\bigsqcap_{j\in J^{i-1}} \mathrm{SP}_j(x_n)\right|.$$

Let s be the number of construction steps needed, i.e. $J := J^s$ is such that property (1) of the lemma holds for this subset of $I(x_n)$. By definition of S_n, it is clear that such a subset exists because $I(x_n)$ itself satisfies property (1).
For each construction step i, we let

$$\Gamma_i := \{\overline{\sigma} \in \bigtimes_{j \in J^i} \mathrm{Sym}(\mathrm{SP}_j(x_n)) \mid \text{ there is a } \pi \in \mathrm{Sym}_n \text{ that realises } \overline{\sigma}\}.$$

Furthermore, for each step i we let k_i be the increase in the number of parts in the intersection that is achieved in this step:

$$k_i := |\bigsqcap_{j \in J^i} \mathrm{SP}_j(x_n)| - |\bigsqcap_{j \in J^{i-1}} \mathrm{SP}_j(x_n)|.$$

The main part of the proof consists in showing the following

▷ Claim 31. For each step i, the size of $|\Gamma_i|$ is bounded by

$$|\Gamma_i| \leq \prod_{j=1}^{i} (\min\{(k_j + 1), f(n)\})!$$

Proof. Via induction on i. For $i = 1$, we have $k_1 = |\mathrm{SP}_{j_1}(x_n)| \leq f(n)$, where j_1 is the level chosen in the first step of the construction of J. The group Γ_1 is a subgroup of $\mathrm{Sym}(\mathrm{SP}_{j_1}(x_n))$, whose size is bounded by $|\mathrm{SP}_{j_1}(x_n)|!$. Therefore, the claim holds. For the inductive step, consider the step $i + 1$ of the construction. Let j_{i+1} be the level that is added in this step. In order to bound the size of Γ_{i+1}, we consider for each $\overline{\sigma} \in \Gamma_i$ the following set:

$$\Gamma_{i+1}^{\overline{\sigma}} := \{\sigma \in \mathrm{Sym}(\mathrm{SP}_{j_{i+1}}(x_n)) \mid \text{there is a } \pi \in \mathrm{Sym}_n \text{ that realises } \sigma \text{ and } \overline{\sigma}\}.$$

We need to show that for each $\overline{\sigma} \in \Gamma_i$, it holds $|\Gamma_{i+1}^{\overline{\sigma}}| \leq (\min\{(k_{i+1} + 1), f(n)\})!$.
Since $|\mathrm{SP}_{j_{i+1}}(x_n)| \leq f(n)$, the bound $|\Gamma_{i+1}^{\overline{\sigma}}| \leq f(n)!$ is clear. It remains to show that for an arbitrary fixed $\overline{\sigma} \in \Gamma_i$, it holds $|\Gamma_{i+1}^{\overline{\sigma}}| \leq (k_{i+1} + 1)!$.
For a part $P \in \mathrm{SP}_{j_{i+1}}(x_n)$, let

$$\mathbf{Q}(P) := \{Q \in \bigsqcap_{j \in J^i} \mathrm{SP}_j(x_n) \mid Q \cap P \neq \emptyset\}.$$

We define an equivalence relation $\sim \subseteq (\mathrm{SP}_{j_{i+1}}(x_n))^2$: For parts $P, P' \in \mathrm{SP}_{j_{i+1}}(x_n)$ we let

$$P \sim P' \text{ iff } \mathbf{Q}(P) = \mathbf{Q}(P').$$

The images of each part $\mathrm{SP}_{j_{i+1}}(x_n)$ under permutations in $\Gamma_{i+1}^{\overline{\sigma}}$ are contained in a single equivalence class of $\sim$: Every $\pi \in \mathrm{Sym}_n$ that realises any $\sigma \in \Gamma_{i+1}^{\overline{\sigma}}$ also realises $\overline{\sigma} \in \Gamma_i$. Hence, by Lemma 16, all such π induce the same $\vartheta_{\overline{\sigma}} \in \mathrm{Sym}(\bigsqcap_{j \in J^i} \mathrm{SP}_j(x_n))$. This means that for any $\sigma \in \Gamma_{i+1}^{\overline{\sigma}}$, and every part $P \in \mathrm{SP}_{j_{i+1}}(x_n)$, $Q \in \bigsqcap_{j \in J^i} \mathrm{SP}_j(x_n)$,

$$\sigma(P) \cap \vartheta_{\overline{\sigma}}(Q) \neq \emptyset \text{ iff } Q \in \mathbf{Q}(P).$$

Therefore, all possible images $\sigma(P) \in \mathrm{SP}_{j_{i+1}}(x_n)$, for all $\sigma \in \Gamma_{i+1}^{\overline{\sigma}}$ must be in the same equivalence class of $\sim$. Consequently, we can bound $|\Gamma_{i+1}^{\overline{\sigma}}|$ as follows: Let m be the number of equivalence classes of $\sim$ and let $\ell_1, .., \ell_m$ denote the sizes of the respective classes. Then from our observations so far it follows:

$$|\Gamma_{i+1}^{\overline{\sigma}}| \leq \prod_{t \in [m]} \ell_t! \qquad (\star)$$

Next, we establish a relationship between the properties of $\sim$ and the number k_{i+1}:

$$\begin{aligned} k_{i+1} &= \Big|\mathrm{SP}_{j_{i+1}}(x_n) \sqcap \prod_{j\in J^i} \mathrm{SP}_j(x_n)\Big| - \Big|\prod_{j\in J^i} \mathrm{SP}_j(x_n)\Big| \\ &= \sum_{Q\in\prod_{j\in J^i}\mathrm{SP}_j(x_n)} (|\{P \in \mathrm{SP}_{j_{i+1}}(x_n) \mid Q \in \mathbf{Q}(P)\}| - 1) \\ &\geq \sum_{P\in\mathrm{SP}^{[P]_\sim,}_{j_{i+1}}(x_n)} (|[P]_\sim| - 1) \\ &= |\mathrm{SP}_{j_{i+1}}(x_n)| - m \end{aligned}$$

The first equality is due to the fact that each part $Q \in \prod_{j\in J^i}$ is split into as many parts as there are parts in $\mathrm{SP}_{j_{i+1}}(x_n)$ intersecting Q.
To see why the inequality holds, fix a choice function g that maps each equivalence class $[P]_\sim$ to a part $Q \in \mathbf{Q}(P)$. By definition of $\sim$, we have $g([P]_\sim) \in \mathbf{Q}(P')$ for every $P' \in [P]_\sim$. Hence, for every $Q \in \prod_{j\in J^i} \mathrm{SP}_j(x_n)$, it holds:
$|\{P \in \mathrm{SP}_{j_{i+1}}(x_n) \mid Q \in \mathbf{Q}(P)\}| - 1 \geq \sum_{[P]_\sim\in g^{-1}(Q)}(|[P]_\sim| - 1)$.
We can sum up the result of these considerations like this:

$$m \geq |\mathrm{SP}_{j_{i+1}}(x_n)| - k_{i+1}. \tag{$\star\star$}$$

Let us now finish the proof of the claim:

We have already established the upper bound $(\star)$ for $|\Gamma^{\overline{\sigma}}_{i+1}|$. Let $p \in [m]$ be such that $\ell_p \geq \ell_t$ for all $t \in [m]$. A consequence of $(\star\star)$ is: $\ell_p \leq k_{i+1} + 1$. It can be checked that the values $\ell_1, ...\ell_m$ that maximise the bound in $(\star)$ and satisfy $(\star\star)$ are such that $\ell_t = 1$ for all $t \neq p$. Therefore, $(\star)$ becomes:

$$|\Gamma^{\overline{\sigma}}_{i+1}| \leq \ell_p! \leq (k_{i+1} + 1)!$$

This concludes the proof of the claim. ◁
Hence, in order to finish the proof of the lemma, we have to bound

$$|\Gamma_s| \leq \prod_{i=1}^{s} (\min\{(k_i + 1), f(n)\})!$$

from above (recall that s is the number of steps needed to construct J satisfying property (1)). We know that $\sum_{i=1}^{s} k_i$ is some fixed value $\leq n$. The value of the above product and the sum solely depends on the sequence $(k_i)_{i\in[s]}$. One can see by a "redistribute-weight argument" that the value of the product is maximised for a sequence $(k_i)_{i\in[s]}$, where every k_i is either 1 or $k_i = f(n) - 1$ (and there may be exactly one k_i with $1 < k_i < f(n) - 1$). For such a sequence of k_is, the value of the product is at most

$$|\Gamma_s| \leq \prod_{i=1}^{s} (\min\{(k_i + 1), f(n)\})! \leq f(n)!^{n/(f(n)-1)} \cdot 2^n$$ ◀

8.2 Proof of Lemma 27

For the proof of Lemma 27, we make use of the following small observation.

▶ **Lemma 32.** *Let $B \subseteq \{0,1\}^n$. The partition $\bigsqcap B = \bigsqcap_{b\in B} SP(b)$ is a supporting partition for B.*

Proof. By the definition of the intersection, every string $b \in B_n$ is constant on every part $P \in \bigsqcap B$. Hence, $\mathrm{Stab}^\bullet_n(\bigsqcap B) \subseteq \mathrm{Stab}_n(B_n)$. This is the definition of a supporting partition. ◀

▶ **Lemma 27.** *There exists a subset $A \subseteq B_n$ of size $|A| \leq \frac{|S_n|}{2}$ such that for each part $P \in \bigsqcap A$, one of the following two statements is true:*

1. *$|P \cap S_n| \leq 2$; **or:***
2. *$|P \cap S_n| > 2$ and for every $b \in B_n \setminus A$, one of these two conditions holds:*
   - *b is constant on $P \cap S_n$; **or***
   - *$b[P \cap S_n]$ is* imbalanced *and, for every $P' \in \bigsqcap A$ with $P' \neq P$, $|P' \cap S_n| > 2$, b is constant on $P' \cap S_n$.*

By $b[P \cap S_n]$ we mean the substring of b at the positions in $P \cap S_n$, and being imbalanced *means that $b[P \cap S_n]$ contains exactly one 0 and there is a 1 at all other positions, or vice versa (exactly one 1 and the rest 0).*

Proof. We construct A stepwise, starting with $A^0 := \emptyset$, and adding one string $a_i \in B_n$ in step i. For step $i+1$ of the construction, assume we have constructed A^i. For $k \in [n]$, we write $P_i(k)$ for the part of $\bigsqcap A^i = \bigsqcap_{a \in A^i} \mathrm{SP}(a)$ that k is in. Now we let

$$K_i := \{k \in S_n \mid |P_i(k) \cap S_n| > 2\}.$$

This is the set of positions whose parts need to be refined more. If $K_i = \emptyset$, then the construction is finished because all parts of $\bigsqcap A^i$ satisfy condition 1 of the lemma. So assume $K_i \neq \emptyset$. By Lemma 32, $\bigsqcap B_n$ is a supporting partition for B_n and therefore at most as coarse as $\mathrm{SP}(B_n)$. Hence, all positions in S_n are in singleton parts of $\bigsqcap B_n$.
We conclude that for all $k \in K_i$, there must be a string $b \in B_n \setminus A^i$ that can be added to A^i in order to make $P_i(k) \cap S_n$ smaller when it is intersected with $\mathrm{SP}(b)$. In fact, there may be several such strings b that we could choose to add in this step of the construction. So let

$$C_k := \{b \in B_n \setminus A^i \mid b \text{ is non-constant on } P_i(k) \cap S_n\}$$

be the non-empty set of such candidate strings. We restrict our candidate set further:

$$\begin{aligned}\widehat{C}_k := \{&b \in C_k \mid \text{there are two distinct parts } P, P' \in \textstyle\bigsqcap A^i \\ &\text{s.t. } b \text{ is non-constant on } P \cap S_n \text{ and } P' \cap S_n, \text{ and} \\ &|P \cap S_n| > 2 \text{ and } |P' \cap S_n| > 2\} \\ \cup \{&b \in C_k \mid b[P_i(k) \cap S_n] \text{ is not imbalanced}\}.\end{aligned}$$

We pick our next string a_{i+1} that is added in this step of the construction from one of the sets $\widehat{C}_k$, where k ranges over all positions in K_i. If $\widehat{C}_k = \emptyset$ for all these k, then A^i is already the desired set A because it satisfies the conditions of the lemma.
Otherwise, we choose a_{i+1} arbitrarily from one of the $\widehat{C}_k$ and set $A^{i+1} := A^i \cup \{a_{i+1}\}$. Then we proceed with the construction until $K_i = \emptyset$ or all $\widehat{C}_k$ are empty. In both cases, the constructed set is as required by the lemma.
It remains to show: $|A| \leq \frac{|S_n|}{2}$, i.e. that the construction process consists of at most $\frac{|S_n|}{2}$ steps. We do this by defining a potential function Φ that associates with any partition $\mathcal{P}$ of $[n]$ a natural number $\leq n$ that roughly says how many further refinement steps of $\mathcal{P}$ are at most possible. Concretely:

$$\Phi(\mathcal{P}) := \sum_{P \in \mathcal{P}} \max\{(|P \cap S_n| - 2), 0\}.$$

If $\mathcal{P}$ contains as its only part the whole set $[n]$, then $\Phi(\mathcal{P}) = |S_n| - 2$. Now observe that a necessary condition for adding a new string a_{i+1} to A is the existence of a part P with $|P \cap S_n| > 2$ in the current partition $\mathcal{P} = \bigsqcap A^i$. This is the case if and only if $\Phi(\mathcal{P}) > 0$. Therefore, all that remains to show is:

$$\Phi\left(\bigsqcap A^i\right) - \Phi\left(\bigsqcap A^{i+1}\right) \geq 2 \tag{$\star$}$$

for all construction steps i. Consider step $i+1$: We add $a_{i+1} \in \widehat{C}_k$ (for some $k \in K_i$) to A^i_n. It can be checked that the definition of $\widehat{C}_k$ ensures that $(\star)$ holds for $\bigsqcap A^i$ and $\bigsqcap A^{i+1} = \bigsqcap A^i \sqcap a_{i+1}$: The new string a_{i+1} either splits two distinct parts, or, if it only splits one part, it splits it into two parts of size at least two. ◀

8.3 Proof of Lemma 28

▶ **Lemma 28.** *Let $A \subseteq B_n$ be the subset that exists by Lemma 27, and let $p : A \longrightarrow B_n$ be an injective function. Let*

$$\Gamma_p := \{\pi \in Stab_n(B_n) \mid p(a)^\pi = a \text{ for all } a \in A\}.$$

Further, let

$$P_{>2} := \{k \in S_n \mid |P(k) \cap S_n| > 2, \text{ where } P(k) \in \bigsqcap A \text{ is the part that } k \text{ is in}\}.$$

Then for any $\pi, \pi' \in \Gamma_p$ such that $\pi^{-1}|_{([n]\setminus P_{>2})} = \pi'^{-1}|_{([n]\setminus P_{>2})}$, it also holds $\pi^{-1}|_{P_{>2}} = \pi'^{-1}|_{P_{>2}}$.

Proof. For a contradiction, we assume that there exist $\pi, \pi' \in \Gamma_p$ such that $\pi^{-1}|_{([n]\setminus P_{>2})} = \pi'^{-1}|_{([n]\setminus P_{>2})}$, but $\pi^{-1}|_{P_{>2}} \neq \pi'^{-1}|_{P_{>2}}$. Then there is $x \in [n]$ such that $\pi(x) \in P_{>2}$, and $\pi'(x) \neq \pi(x)$ (i.e. $\pi(x)$ is the point where π^{-1} and π'^{-1} differ). Let $y := \pi(x), y' := \pi'(x)$. Let $P(y) \in \bigsqcap A$ be the part that y is in, and let $\widehat{P}(y) := P(y) \cap S_n$. We know that $y' \in P(y)$, too, because $\pi, \pi' \in \Gamma_p$, so this follows from Lemma 25. As $y \in P_{>2}$, in particular, $y \in S_n$. Hence, also $x, y' \in S_n$ because $\pi, \pi' \in \mathrm{Stab}_n(B_n)$, and by Lemma 11, singleton parts must be mapped to singleton parts. We conclude that we even have $y' \in \widehat{P}(y)$.
Now, our goal is to show that the transposition $\tau := (y\ y')$ is contained in $\mathrm{Stab}_n(B_n)$. This is a contradiction because in $\mathrm{SP}(B_n)$, y, y' are both in singleton parts, but if $\tau \in \mathrm{Stab}_n(B_n)$, then the coarsest supporting partition $\mathrm{SP}(B_n)$ can be coarsened to another supporting partition by letting $\{y, y'\}$ be one single part.
In order to show $\tau \in \mathrm{Stab}(B_n)$, we only need to deal with those strings in B_n which are not constant on the positions $\{y, y'\}$. More precisely, we have to show that every $b \in B_n$ with $b_y \neq b_{y'}$ has a "swapping partner" $b' \in B_n$ where $b'_y = b_{y'}$ and vice versa, and $b'_i = b_i$ for all other i.
So take any $b \in B_n$ such that w.l.o.g. $b_y = 0, b_{y'} = 1$. Note that $b \notin A$, as every string in A is constant on $P(y)$ (otherwise, $P(y)$ would not be a single part in $\bigsqcap A$). Furthermore, $|\widehat{P}(y)| > 2$, since $y \in P_{>2}$. Therefore, Lemma 27 implies that the substring $b[\widehat{P}(y)]$ is imbalanced and b is constant on every $P' \cap S_n$, for all $P' \in \bigsqcap A$ with $P' \neq P(y)$, $|P' \cap S_n| > 2$. W.l.o.g. let the imbalance of $b[\widehat{P}(y)]$ be such that $b_i = 1$ for every position $i \in \widehat{P}(y), i \neq y$. We claim that $b' := b^{\pi' \circ \pi^{-1}}$ is the desired swapping partner of b, i.e. $b^\tau = b'$ and vice versa. Note that $b' \in B_n$ because π, π' stabilise the set B_n.
To see that $b' = b^\tau$, consider firstly $b^{\pi^{-1}} \in B_n$. Obviously, $(b^{\pi^{-1}})_x = 0$. Hence, $(b^{\pi' \circ \pi^{-1}})_{y'} = 0$. Moreover, the substring $b^{\pi^{-1}}[\pi^{-1}(\widehat{P}(y))]$ is imbalanced just like $b[\widehat{P}(y)]$, so $(b^{\pi^{-1}})_j = 1$ for all $j \in \pi^{-1}(\widehat{P}(y)) \setminus \{x\}$. As a consequence of Corollary 26, we have $\pi^{-1}(\widehat{P}(y)) = \pi'^{-1}(\widehat{P}(y))$,

so $(\pi' \circ \pi^{-1})(\widehat{P}(y)) = \widehat{P}(y)$. Therefore, the substring $b^{\pi' \circ \pi^{-1}}[\widehat{P}(y)]$ is also imbalanced and has a 1 at each position except y'.
This shows that $(b^\tau)[\widehat{P}(y)] = b'[\widehat{P}(y)]$. It remains to show that $b_i = b'_i$ for all $i \in [n] \setminus \widehat{P}(y)$. We have $(\pi' \circ \pi^{-1})(i) = i$ for $i \in [n] \setminus P_{>2}$, because $\pi^{-1}|_{([n]\setminus P_{>2})} = \pi'^{-1}|_{([n]\setminus P_{>2})}$, so $b_i = b'_i$ for $i \in [n] \setminus P_{>2}$.
For $i \in P_{>2} \setminus \widehat{P}(y)$, let $\widehat{P}(i)$ be the part of $\bigsqcap A$ that i is in, intersected with S_n. As already said, we know from Lemma 27 that b is constant on $\widehat{P}(i)$. Analogously to what we argued already for $\widehat{P}(y)$, we get that $(\pi' \circ \pi^{-1})(\widehat{P}(i)) = \widehat{P}(i)$, so also for $i \in P_{>2} \setminus \widehat{P}(y)$, we have $b_i = b'_i$.
In total, this shows that indeed, $b' = b^\tau$, and since $b' \in B_n$, we have $\tau \in \mathrm{Stab}_n(B_n)$. This is a contradiction and finishes the proof of the lemma. ◀