On the Lattice of Clones Below the Polynomial Time Functions

Elmar Böhler

Universität Würzburg
Lehrstuhl für Theoretische Informatik
Am Hubland, 97230 Würzburg, Germany
boehler@informatik.uni-wuerzburg.de

Abstract. A clone is a set of functions that is closed under generalized substitution. The set FP of functions being computable deterministically in polynomial time is such a clone. It is well-known that the set of subclones of every clone forms a lattice. We study the lattice below FP, which contains other important function complexity classes like FL and FNC^i. We show that the lattice is dualatomic and determine some of its dualatoms. We show that no time-complexity class can be a dualatom in this lattice. We show that there are uncountably many subclones of FP.

1 Introduction

Computer science is an inductive discipline. The idea of the algorithm, that stands in the very core of computer science, is the idea of using basic primitives to describe and analyze complex procedures. The idea is global, independently of whether you are in functional programming, languages, or Turing machines. Therefore, the notion of closures is natural: You are provided with a small set of primitives and operations, and everything that can be build by applying the operations on the primitives, belongs to the closure.

At first sight, complexity theory seems to define its objects the other way round. Many important complexity classes like P or PSPACE are defined by resource bounds. So P was not first defined as the class of sets that can be built from a small set of primitives using a few basic operations but as a class of sets, all sharing a certain property: Their decidability in polynomial time. One of the most difficult problems is to seperate these complexity classes. For P and PSPACE, for example, this could obviously be done by finding a problem that is in $\mathrm{PSPACE} \setminus \mathrm{P}$. However, such problems are often very hard to find and the closest concept we have to date, is that of complete sets.

Completeness of a set for a complexity class is always in respect to some reducibility. Here we find the concept of closures once again. For every complexity class, you can build the closure in respect to some reducibility. So, for example, if $\mathrm{P} \neq \mathrm{NP}$ then there are infinitely many classes that are closed under the $\leq_m^p$-reducibility (so-called *degrees*) between P and NP. Moreover, they form an upper semilattice in which every countable partial order can be embedded as suborder [Lad75,Meh76].

From a certain point of view, P is the most natural $\leq_m^p$-degree, because the reducing functions are drawn from its function-counterpart FP. Moreover, FP itself is closed under substitution and is thus an example for a function algebra.

In that algebra the primitives are functions and the operations are those you need to build polynoms or formulas over the functions, in order to create new ones. These operations are called *superposition* and classes of functions being closed under superposition are called *clones*. Universal algebra teaches us, that the set of subclones to one specific clone form a complete lattice [Coh65,Grä79].

The lattice below FP is the object we want to study in this paper. It is not difficult to see that other very interesting complexity classes form subclones of FP: For example, FL the class of functions computable in logarithmic space. Or the classes FNC^i which are the function analogons of those sets which are decidable by Boolean circuits of polynomial size and polylogarithmic depth. These classes are known to be in FP, but it is not clear whether this inclusion is proper. If, for example $\mathrm{FL} = \mathrm{FP}$ would hold, then the lattice of clones below FL would have to be the same than that below FP. We show that the lattice below FP is dualatomic, i.e. for every infinte chain $C_1 \subset C_2 \subset \cdots \subset \mathrm{FP}$ of clones below FP there is a proper subclone (called *dualatom*) $B \subset \mathrm{FP}$ that contains the chain. So, for example, for FL there are only three possibilities: First, $\mathrm{FL} = \mathrm{FP}$. Then FL has the same dualatoms as FP. Secondly, FL is a dualatom of FP. Then FL is very close to FP, "just one function away", since for all $f \in \mathrm{FP} \setminus \mathrm{FL}$ is the superposition-closure of $\{f\} \cup \mathrm{FL}$ already equal to FP. Finally, FL could already be a proper subset of a dualatom of FP. Perhaps it is easier to show such an inclusion?

The property of FP to be dualatomic stems from the fact, that FP can be finitely generated. Call FNC the union of all FNC^i. Then FNC is a clone, also. It is an open question whether $\mathrm{FNC} = \mathrm{FP}$; this question is often set equal to the question of whether all functions from FP can be efficiently parallelized. But if $\mathrm{FP} = \mathrm{FNC}$ would hold, then FNC would have to be finitely generated, too, and as immediate consequence of that, the NC-hierarchy would collapse.

The algebraic approach to complexity classes is not new at all. Various complexity classes have been characterized using a small set of functions and some basic algebraic operators [Wag86,VW96,Clo99]. However, often these operators were specially designed for the various complexity classes. The difference in this approach is that we analyze the algebraic structure of complexity classes that can can be build with one very natural set of operators.

After the preliminaries, we show that the clone lattice below FP is dualatomic in subsection 3.1. Then we identify infinitely many of the dualatoms in subsection 3.2 and show that there can be no time-complexity class that is a dualatom that lattice. In the last subsection 3.3 we show that the lattice, although it is only countably deep, has uncountably many elements.

2 Preliminaries

We begin by introducing the superposition operators.

Definition 1. *Let* $\mathrm{I}_1^2(x, y) =_{\mathrm{df}} x$ *for all* $x \in \mathbb{N}$. *Let* f, g, h *be functions of arity* n, m, ℓ *respectively, where* $n \geq 1$ *and* $\ell \geq 2$.

- $\mathrm{ZV}(h)(x_1, \ldots, x_n) =_{\mathrm{df}} h(x_2, \ldots, x_n, x_1)$
- $\mathrm{LV}(h)(x_1, \ldots, x_n) =_{\mathrm{df}} h(x_1, \ldots, x_n, x_{n-1})$
- $\mathrm{ID}(h)(x_1, \ldots, x_n) =_{\mathrm{df}} h(x_1, \ldots, x_{n-1}, x_{n-1})$
- $\mathrm{SB}(f, g)(x_1, \ldots, x_{n-1}, y_1, \ldots, y_m) =_{\mathrm{df}} f(x_1, \ldots, x_{n-1}, g(y_1, \ldots, y_m))$

A set of functions that contains I_1^2 *and is closed under* $\mathrm{ZV}, \mathrm{LV}, \mathrm{ID}, \mathrm{SB}$ *is closed under* superposition *and is called* clone. *For a set of functions* B*, let* $[B]$ *be the superposition closure of* B*.*

We remark, that the operations ZV and LV can be used to generate arbitrary permuations of the variables. More intuitively, $[B]$ is the set of functions that can be described with formulas over B, i.e. formulas with connectors representing functions from B. For example, if $B = \{f, g\}$, where f is 3-ary and g is 2-ary, then $f(x, g(y, z), y)$ and $g(x, x)$ and x describe functions from $[B]$. Since the identity function $\mathrm{id}(x) = x$ can be represented by the formula "x", which contains no function symbols at all, all clones always contain the identity functions. A function f is an identity function if $f(x_1, \ldots, x_n) = x_i$ for an $i \in \{1, \ldots, n\}$. Let I be the set of all identity functions. We include the function I_1^2 instead of id, because we want to make possible the introduction of *fictive* variables, i.e. variables which have no influence on the value of a function. To be more precisely, x_1 is a fictive variable in $f(x_1, \ldots, x_n)$, if for all $a, b, a_2, \ldots, a_n \in \mathbb{N}$ holds $f(a, a_2, \ldots, a_n) = f(b, a_2, \ldots, a_n)$. Of course, the most powerful operator of superposition is the substitution operator. Therefore in many cases it suffices to show that a set B is closed under substitution in order to show that it is closed under superposition. For a clone A, the set $\{B : B \text{ is clone and } B \subseteq A\}$ forms a lattice with respect to the operations $\cap, \sqcup$ where $\cap$ is the normal intersection of sets and $C \sqcup D =_{\mathrm{df}} [C \cup D]$. We denote this lattice by $\mathcal{L}(A)$. The least element of $\mathcal{L}(A)$ is always I. If $B \in \mathcal{L}(A)$, we say B is *subclone* of A and A is a *superclone* of B.

A clone A is called *finitely generated*, if there is a finite set of functions B, such that $[B] = A$. In this case, B is called a *base* of A. A subclone of $B \in \mathcal{L}(A)$ is called *dualatom* of A or *precomplete for* A, if $B \neq A$ and there is no $C \in \mathcal{L}(A)$ such that $B \subset C \subset A$. The lattice is called *dualatomic* if for all $C \in \mathcal{L}(A)$ we have $C \subseteq B$ for a dualatom B of A. In other words, if such a lattice contains an infinite chain of the form $C_1 \subset C_2 \subset \cdots \subset A$ then there is a dualatom B of A such that $C_1 \subset C_2 \subset \cdots \subset B$.

Proposition 1 ([Neu37]). *If* A *is a finitely generated clone, then* $\mathcal{L}(A)$ *is dualatomic.*

Proof. Suppose $\mathcal{L}(A)$ were not dualatomic. Then there would be subclones $C_1 \subset C_2 \subset \cdots \subset A$. Let $C =_{\mathrm{df}} \bigcup_i C_i$. Obviously, $[C]$ is a proper superclone of all C_i's and since $\mathcal{L}(A)$ is not dualatomic, $[C] = A$. Since A is finitely generated, there is a finite B with $[B] = A$. Because of the nature of C there must be an i such that $B \subseteq C_i$. But then, $C_i = A$.

Throughout this paper, we will need some special functions. Let $\mathrm{bin} : \mathbb{N} \to \{0,1\}^*$ be the usual binary encoding of natural numbers without leading zeroes. For an $x \in \mathbb{N}$, let $|x|$ be the number of digits in $\mathrm{bin}(x)$, i.e. $|x| =_{\mathrm{df}} \lfloor \log_2 x + 1 \rfloor$ if $x > 0$ and $|0| =_{\mathrm{df}} 1$. Let $\mathrm{dbin}(x) =_{\mathrm{df}} a_1 a_1 a_2 a_2 \ldots a_n a_n$, if $\mathrm{bin}(x) = a_1 \ldots a_n$ and let $\mathrm{pair}(x, y) =_{\mathrm{df}} \mathrm{dbin}(x) 01 \mathrm{dbin}(y)$. Let $\mathrm{succ}(x) =_{\mathrm{df}} x + 1$, $\mathrm{pad}(x) =_{\mathrm{df}} 1^{|x|^2}$, and $\mathrm{c}_j(x) =_{\mathrm{df}} j$ for all $j \in \mathbb{N}$.

Let $\mathcal{K} \subseteq \mathbb{N}^{\mathbb{N}}$ be a set of functions. We say $f \in \mathrm{FDTIME}(\mathcal{K})$ if there is a $t \in \mathcal{K}$ and a Turing machine with a fixed number of tapes that computes $f(x)$ for all $x \in \mathbb{N}$ using at most $t(|x|)$ steps. Let $M_1, M_2, \ldots$ be an enumeration of turing machines such that M_1 calculates $\mathrm{succ}(x)$ in time $2|x| + 2$ and $g : \mathbb{N} \to \mathbb{N}$ with

$g(i) = M_i$ is in $\mathrm{FDTIME}(O(n \log n))$. Furthermore, we want the enumeration such that a universal machine is able to simulate a step of M_i in time $|i|$.
Let

$$\mathrm{univ}(i, x) =_{\mathrm{df}} \begin{cases} M_i \text{ on input } x & \text{, if } M_i \text{ holds after } 2(|i||x|^2 + |i|) \text{ steps} \\ 1^{|x|^2} & \text{otherwise.} \end{cases}$$

Note that if a function f is in $\mathrm{DTIME}(O(n^2))$ then there is an i such that M_i computes f in at most $2(|i||x|^2 + |i|)$ steps. Observe, that $\mathrm{pair}, \mathrm{succ}, c_j \in \mathrm{FDTIME}(O(n))$ for all $j \in \mathbb{N}$. Let $\mathrm{FP} =_{\mathrm{df}} \mathrm{FDTIME}(O(n^{O(1)}))$ be the set of functions that can be computed deterministically in polynomial time.

3 On the Nature of the Clone Lattice Below FP

It is not difficult to verify, that FP is closed under superposition and hence is a clone. The lattice $\mathcal{L}(\mathrm{FP})$ is the object we want to study in this paper. We show that it is dualatomic and we will determine some of its dualatoms. We show, that no complexity class of the type $\mathrm{DTIME}(\mathcal{K})$ is a dualatom of FP, where $\mathcal{K}$ is an arbitrary set of functions. Finally, we will look at the size of the lattice.

3.1 The Lattice is Dualatomic

We show that FP is finitely generated by explicitely giving a finite base for it. This base contains mainly a universal machine, i.e. a machine that is able to interpret encodings of other machines and simulate those. We have to make sure that the running time of this universal machine does not exceed polynomial time for one fixed polynomial. Then we use a padding argument in order to carry out any calculation that can be done in polynomial time for an arbitrary polynomial.

Theorem 1. *The set* $\{\mathrm{univ}, \mathrm{pair}, \mathrm{c}_1\}$ *is a base of* FP.

Proof. Obviously, $\mathrm{univ}, \mathrm{pair}$ and c_1 are in FP and since FP is closed under superposition, for $B =_{\mathrm{df}} \{\mathrm{univ}, \mathrm{pair}, c_1\}$ we have $[B] \subseteq \mathrm{FP}$. We show that $[B] \supseteq \mathrm{FP}$. Note that $\mathrm{succ}(x) = \mathrm{univ}(c_1(x), x)$ and therefore $\mathrm{succ} \in [B]$. Since we have the successor function and the constant 1 in our closure it is obvious, that for all $i > 1$ the function c_i is in $[B]$. Let i be the number of a machine that runs for more than $2(|i||x|^2 + |i|)$ steps for all $x \in \mathbb{N}$. Then $\mathrm{pad}(x) = \mathrm{univ}(c_i(x), x)$ because we always fall into the second option of the case distinction of the definition of univ. Hence $\mathrm{pad} \in [B]$. Now let $s : \mathbb{N}^m \to \mathbb{N}$ be in FP. There is an $s' : \mathbb{N} \to \mathbb{N}$ such that $s'(\mathrm{pair}(x_1, \mathrm{pair}(x_2, \ldots, \mathrm{pair}(x_{m-1}, x_m) \ldots))) = s(x_1, \ldots, x_m)$ and $s' \in \mathrm{FP}$. Hence there are $i, k \in \mathbb{N}$ such that M_i calculates s' in time $k + kn^k$. Let

$$s''(x) =_{\mathrm{df}} \begin{cases} s'(y) & \text{, if } x = 1^z 01\mathrm{dbin}(y)\text{, where } z \geq k + k|y|^k \\ 0 & \text{otherwise} \end{cases}$$

Then $s'' \in \mathrm{FP}$ and there is a $\ell \in \mathbb{N}$ such that M_ℓ computes s'' in time $O(n)$. Hence

$$\begin{aligned} s'(x) = \mathrm{univ}\ (\ & c_\ell(x), \\ & \mathrm{pair}(\ \underbrace{\mathrm{pad}(\mathrm{pad}(\ldots \mathrm{pad}}_{\lceil \log k \rceil}(x) \ldots))\ ,\ x) \\ &) \end{aligned}$$

is in $[B]$ and therefore $s \in [B]$.

Corollary 1. *The lattice of subclones of* FP *is dualatomic.*

We have seen, that FP can be generated by only three functions, all of wich are contained in $\mathrm{DTIME}(O(n^4))$. The critical point in the proof of Theorem 1 is that we are able to pad our inputs to an arbitrary polynomial length and that we can do that with only a constant number of substitutions of pad. In our proof, we use a padding function that generates outputs of quadratic length. It is easy to see that a similar argumentation holds if we can generate outputs of length $n^{1+\varepsilon}$ for an $\varepsilon > 0$.

Proposition 2. *Let $\mathcal{K}$ be a class of function such that for an $\varepsilon > 0$ a function $t(n) \geq n^{1+\varepsilon}$ is in $\mathcal{K}$. Then* $\mathrm{FP} \subseteq [\mathrm{DTIME}(\mathcal{K})]$.

3.2 Dualatoms

So, since $\mathcal{L}(\mathrm{FP})$ is dualatomic, it would be very nice to know the dualatoms. The dualatoms we identify in this section are very close to FP. It is possible, for example, to encode P-complete problems with very little effort in such a way that they can be decided by functions from one of these dualatoms. This gives rise to the assumption, that the structure of $\mathcal{L}(\mathrm{FP})$ is very fine-grained.
In the following we describe Boolean functions using propositional formulas. We use the connectors $\wedge, \vee, \oplus, \leftrightarrow, \neg, 0$, and 1 for the Boolean functions and, or, xor, equivalence, not, constant zero, and constant one respectively. For $\neg x$, we also write $\overline{x}$ and for $x \wedge y$ we also write xy.
Let BF be the set of all Boolean functions. The lattice $\mathcal{L}(\mathrm{BF})$ of clones of Boolean functions is well known [Pos41,JGK70,BCRV03]. This lattice is dualatomic and has five dualatoms, often referred to as Post's classes. They are defined as follows.

Definition 2.
- $\mathrm{R}_a =_{\mathrm{df}} \{f : f(a, \ldots, a) = a\}$ *for $a \in \{0,1\}$ are the a-*repdroducing *clones.*
- $\mathrm{D} =_{\mathrm{df}} \{f : f(a_1, \ldots, a_n) \neq f(\overline{a_1}, \ldots, \overline{a_n})$ *for all* $a_1, \ldots, a_n \in \{0,1\}\}$ *are the* selfdual *functions.*
- *The set of* monotonic *Boolean functions M is defined* $\mathrm{M} =_{\mathrm{df}} \{f : \alpha, \beta \in \{0,1\}^n, \alpha \leq \beta \rightarrow f(\alpha) \leq f(\beta)\}$ *where for Boolean vectors $\alpha = (a_1, \ldots, a_n), \beta = (b_1, \ldots, b_n)$ holds $\alpha \leq \beta$, if and only if for all $i \in \{1, \ldots, n\}$ holds $a_i \leq b_i$.*
- L *is the set of Boolean functions $f(x_1, \ldots, x_n)$ that can be described by a formula of the form $c_0 \oplus c_1 x_1 \oplus \cdots \oplus c_n x_n$, where $c_i \in \{0,1\}$.*

We generalize the notion of a-reproducing functions.

Definition 3. *For a set $A \subseteq \mathbb{N}$, let $f : \mathbb{N}^n \rightarrow \mathbb{N}$ be called A-reproducing, if for all $a_1, \ldots, a_n \in A$ holds $f(a_1, \ldots, a_n) \in A$. Let R_A be the set of A-reproducing functions.*

Theorem 2. *For all finite $A \subseteq \mathbb{N}$ is $B =_{\mathrm{df}} \mathrm{R}_A \cap \mathrm{FP}$ a dualatom of $\mathcal{L}(\mathrm{FP})$.*

Proof. Obviously, B is a subclone of FP. We show that it is precomplete. For that, let $f \in \mathrm{FP}$ be an m-ary function that is not A-reproducing.
If $A = \{a_0, \ldots, a_{k-1}\}$ and $0 \leq i < k$, let $\pi(a_i) =_{\mathrm{df}} a_{i+1 (\mathrm{mod}\ k)}$. Let $\pi^0(a_i) =_{\mathrm{df}} \pi(a_i)$ and for $j \geq 0$, let $\pi^{j+1}(a_i) =_{\mathrm{df}} \pi(\pi^j(a_i))$. For all $w \in \{0, \ldots, k-1\}^m$ with $w =_{\mathrm{df}} (d_1, \ldots, d_m)$, define $f_w(x) = f(\pi^{d_1}(x), \ldots, \pi^{d_m}(x))$. Observe that all these f_w are in $[B \cup \{f\}]$. Furthermore, if $a \in A$, there is a $w \in \{0, \ldots, k-1\}^m$ such that $f_w(a) \notin A$, since $f \notin \mathrm{R}_A$. Let

$$h'(x_1, \ldots, x_{k^m}) =_{\mathrm{df}} \begin{cases} x_j & \text{, if there is a } 1 \leq j \leq k^m \text{ such that } x_j \notin A \\ x_1 & \text{otherwise} \end{cases}$$

Note, that $h' \in B$. For $1 \leq i \leq k^m$, let $w_i \in \{0, \ldots, k-1\}^m$ such that $w_1 =_{\mathrm{df}} (0, \ldots, 0)$, $w_2 =_{\mathrm{df}} (0, \ldots, 0, 1)$, $\ldots$, $w_{k^m} =_{\mathrm{df}} (k-1, \ldots, k-1)$. Let $h(x) =_{\mathrm{df}} h'(f_{w_1}(x), \ldots, f_{w_{k^m}}(x))$. Then $h \in [B \cup \{f\}]$ and for all $x \in A$ holds $h(x) \notin A$.
Now, let g be an arbitrary n-ary function from FP. Certainly, the function

$$s(x_1, \ldots, x_n, y) =_{\mathrm{df}} \begin{cases} x_1 & \text{, if } x_1, \ldots, x_n, y \in A \\ g(x_1, \ldots, x_n) & \text{otherwise} \end{cases}$$

is in $\mathrm{FP} \cap \mathrm{R}_A$. Then $g(x_1, \ldots, x_n) = s(x_1, \ldots, x_n, h(x_1))$ for all $x_1, \ldots, x_n \in \mathbb{N}$ and therefore $g \in [B \cup \{f\}]$.

For the next dualatoms we take a common computation model for FP-functions and lessen the power of the model slightly. The model is that of Boolean circuits of polynomial size.

Definition 4. *For a Boolean circuit C, let $\mathrm{size}(C)$ be the number of gates of C, and let $\mathrm{depth}(C)$ be the length of the longest path between an input gate of C and an output gate of C. For a set of Boolean functions B and classes of functions $\mathcal{K}_1, \mathcal{K}_2$ on $\mathbb{N}$, let $\mathrm{FSIZE\text{-}DEPTH}_B(\mathcal{K}_1, \mathcal{K}_2)$ be the class of functions computable by logspace-uniform families of circuits with gates from B that are bounded in size and depth by functions from $\mathcal{K}_1$ and $\mathcal{K}_2$. We write $\mathrm{FCPS}_B =_{\mathrm{df}} \mathrm{FSIZE\text{-}DEPTH}_B(O(n^{O(1)}), O(n^{O(1)}))$, for the class of* **f***unctions computable by* **c***ircuits of* **p***olynomial* **s***ize with gates from B.*

For an exact definition of Boolean circuits and logspace-uniform families of circuits, we refer to [Vol99]. For every set of Boolean functions B, for which $[B] = \mathrm{BF}$, we have $\mathrm{FP} = \mathrm{FCPS}_B$, (e.g. $B = \{\mathrm{and}, \mathrm{or}, \mathrm{not}\}$). Note that the actual base B we chose is not relevant: If $[B] = [B']$ then every function from B can be computed by a circuit over B'. Therefore, if $f \in \mathrm{FCPS}_B$ is calculated by the family of circuits $(C_i)_{i \in \mathbb{N}}$ over B, we can find a family of circuits $(C'_i)_{i \in \mathbb{N}}$ over B' by replacing every gate in C_i by the corresponding circuit over B' of constant size. So the circuits in the family $(C'_i)_{i \in \mathbb{N}}$ grow for a constant factor only.
We want to study circuits of polynomial size over sets B of Boolean functions such that $[B] \neq \mathrm{BF}$.

Proposition 3. *If $[B]$ is not a dualatom in $\mathcal{L}(\mathrm{BF})$ then FCPS_B cannot be a dualatom in $\mathcal{L}(\mathrm{FP})$.*

Proof. Let A be a dualatom in $\mathcal{L}(\mathrm{BF})$ with $[B] \subset A$, let A' be a base of A, and let $f \in A \backslash [B]$. Then $f \in \mathrm{FP} \backslash \mathrm{FCPS}_B$, so if FCPS_B were a dualatom in $\mathcal{L}(\mathrm{FP})$ we would expect $[\mathrm{FCPS}_B \cup \{f\}] = \mathrm{FP}$. But since every $g \in \mathrm{FCPS}_B$ can be computed by a family of circuits over B, every $g' \in [\mathrm{FCPS}_B \cup \{f\}]$ can be computed by a family of circuits over $B \cup \{f\}$ and therefore $[\mathrm{FCPS}_B \cup \{f\}] \subseteq \mathrm{FCPS}_{A'} \subset \mathrm{FP}$.

So for FCPS_B to be a dualatom in $\mathcal{L}(\mathrm{FP})$ it is necessary that $[B]$ is a dualatom in $\mathcal{L}(\mathrm{BF})$. However, this condition is not sufficient: For clones of Boolean functions $[A]$ and $[B]$ with $[A] \subset [B] = \mathrm{BF}$, suppose there exists a function $f \in [A]$ that can be computed by relatively small circuits over B, but all circuits over A computing f are huge in comparison. Then $[\mathrm{FCPS}_A \cup \{f\}]$ could be a proper superclone of FCPS_A but would still not contain e.g. the Boolean function $\mathrm{nand} \in \mathrm{FP}$. We will see that this is the case with the monotonic Boolean functions. First, we need to formalize the notion of huge and small circuits with respect to different bases.

Definition 5. *Let B_1 and B_2 be finite sets of Boolean functions. We say B_2* is polynomially unnecessary for B_1 *if for every B_2-circuit C_2 that describes a Boolean function $f \in [B_1]$ there is a polynomial p and a B_1-circuit C_1 describing f with $\mathrm{size}(C_1) \leq p(\mathrm{size}(C_2))$. Otherwise, we call B_2* polynomial necessary for B_1.

Lemma 1. *Let B_1, B_2 be sets of Boolean functions. If $[B_1]$*

1. *is equal to* L*, or*
2. *is equal to* D*, or*
3. *is from* $\{\mathrm{R}_1, \mathrm{R}_0\}$

then B_2 is polynomially unnecessary for B_1.

Proof. The first point is obvious, since every function from L can, by definition, be described by a circuit over a base of L of linear size in the number of relevant variables.

For the second point, let $\mathrm{sd}(x, y, z) =_{\mathrm{df}} x\overline{y} \wedge x\overline{z} \wedge \overline{yz}$. It is known that $\{\mathrm{sd}\}$ is a base for D [Pos41]. Observe, that $\mathrm{sd}(1, x, y) = \mathrm{nand}(x, y)$. Now let f be an n-ary Boolean function from D. Let $f'(x_2, \ldots, x_n) =_{\mathrm{df}} f(1, x_2, \ldots, x_n)$. Since

$$f'(x_2, \ldots, x_n) = f(\mathrm{nand}(x_n, \mathrm{nand}(x_n, x_n)), x_2, \ldots, x_n)$$

there is a $\{\mathrm{nand}\}$-circuit C' computing f' with just two more gates than every $\{\mathrm{nand}\}$-circuit computing f. We build an $\{\mathrm{sd}\}$-circuit C out of C' by replacing every nand-gate g as follows: If y and z are the inputs of g, replace it by an sd-gate with inputs x_1, y, and z. Here, x_1 is an additional input gate and we connect every g with this same gate. Let f_C be the function computed by C. Then $f_C(1, x_2, \ldots, x_n) = f'(x_2, \ldots, x_n) = f(1, x_2, \ldots, x_n)$. On the other hand, since f_C is self-dual, we have $f_C(0, x_2, \ldots, x_n) = \overline{f_C(1, \overline{x_2}, \ldots, \overline{x_n})} = \overline{f'(\overline{x_2}, \ldots, \overline{x_n})} = \overline{f(1, \overline{x_2}, \ldots, \overline{x_n})} = f(0, x_2, \ldots, x_n)$. Hence $f_C = f$.

We show the third point for $B_1 = \mathrm{R}_0$. Obviously $\{\vee, \wedge, \oplus\} \subseteq \mathrm{R}_0$. Let $f \in \mathrm{R}_0$ be described by a circuit C over $\{\wedge, \neg\}$ with inputs $x_1, \ldots, x_n$. We build a new circuit C' by replacing every occurence of a $\neg$ gate in C with an $\oplus$ gate with the following inputs: The first input is the input of the original $\neg$ gate and the second one is $\bigvee_{i=1}^{n} x_i$. Since C' describes an R_0 function, $f_{C'}(0,\ldots,0) = f(0,\ldots,0) = 0$. For every other input, the new $\oplus$ gates behave like $\neg$ gates on their first input, since their second input evaluates to 1. The proof for $B_1 = \mathrm{R}_1$ can be done, by switching 1 and 0, $\wedge$ and $\vee$, and $\oplus$ and $\leftrightarrow$.

In the following proofs, we need two special functions.

- For $k \in \mathbb{N}$, $x = x_1 \ldots x_n \in \{0,1\}^n$, and $i = \min\{k,n\}$ let $\mathrm{id}_k(x) =_{\mathrm{df}} x_i$.
- For $x, y \in \{0,1\}^*$, let $\mathrm{conc}(x,y) =_{\mathrm{df}} xy$ be the function that concats x and y.

Note that for all sets of Boolean functions B, and all $k > 0$, these functions are in FCPS_B.

Theorem 3. *If* $[B] = \mathrm{D}$ *then* FCPS_B *is a precomplete subclone of* FP.

Proof. Obviously $\mathrm{FCPS}_B \subset \mathrm{FP}$. We have to show its precompleteness. For that, let $f \in \mathrm{FP} \setminus \mathrm{FCPS}_B$. Then there is a uniform family of circuits $(C_n^f)_{n\in\mathbb{N}}$ over $\{\vee, \wedge, \neg\}$ that computes f. Also, there is a polynomial p, such that for $i \in \{1, \ldots, p(k)\}$ Boolean functions f_i^k are computed by the i-th output gate of circuit C_k^f. Since $\{\vee, \wedge, \neg\}$ is polynomially unnecessary for $\{\mathrm{sd}\}$, cf. Lemma 1, there must be a $k \in \mathbb{N}$ and an $\ell \in \{1, \ldots, p(k)\}$ such that $f_\ell^k \notin [\{\mathrm{sd}\}]$. That means, that there are $a_1, \ldots, a_k \in \{0,1\}$ such that $f_\ell^k(a_1, \ldots, a_k) = f_\ell^k(\overline{a_1}, \ldots, \overline{a_k})$. For $a \in \{0,1\}$, let $a^1 =_{\mathrm{df}} a$ and $a^0 =_{\mathrm{df}} 1-a$. Observe, that the function $h(x_1 \ldots x_n) =_{\mathrm{df}} x_1^{a_1} \ldots x_1^{a_k}$ is in FCPS_B, in fact, h can be calculated by a family of small circuits over any base that can express negation. Then for all $x \in \mathbb{N}$ holds $|h(x)| = k$ and $f_\ell^k(h(x)) = c$ for some $c \in \{0,1\}$. Therefore there is a $c \in \{0,1\}$ such that the constant function $c(x) = c = \mathrm{id}_\ell(f(h(x)))$ is in FCPS_B.

Assume that $c = 1$. Now let $g \in \mathrm{FP}$ be computed by a uniform family of circuits $(D_n^g)_{n\in\mathbb{N}}$ over $\{\mathrm{nand}\}$ such that each circuit has polynomial size. Regard the family of circuits $(E_n)_{n\in\mathbb{N}}$ where E_1 computes c and E_{n+1} is derived from D_n^g by adding an additional input gate x_0 and replacing every nand gate that has inputs x and y by an sd gate with inputs x_0, x, and y. Then for the function g' that is computed by $(E_n)_{n\in\mathbb{N}}$ and all $x \in \{0,1\}^+$ holds $g'(1x) = g(x)$. Hence $g(x) = g'(\mathrm{conc}(c(x), x))$ is in FCPS_B.

If $c = 0$ the proof is analogous with a family of circuits over $\{\mathrm{nor}\}$.

Theorem 4. *If* $[B] \in \{\mathrm{R}_0, \mathrm{R}_1\}$ *then* FCPS_B *is a precomplete subclone of* FP.

Proof. We prove the claim for $[B] = \mathrm{R}_0$. Obviously $\mathrm{FCPS}_B \subset \mathrm{FP}$. We have to show its precompleteness. For that, let $f \in \mathrm{FP} \setminus \mathrm{FCPS}_B$. Then there is a uniform family of circuits $(C_n^f)_{n\in\mathbb{N}}$ over $\{\vee, \wedge, \neg\}$ that computes f. Also, there is a polynomial p, such that for $i \in \{1, \ldots, p(k)\}$ Boolean functions f_i^k are computed by the i-th output gate of circuit C_k^f. Since $\{\vee, \wedge, \neg\}$ is polynomially unnecessary for $\{\vee, \wedge, \oplus\}$, cf. Lemma 1, there must be a $k \in \mathbb{N}$ and an $\ell \in \{1, \ldots, p(k)\}$ such that $f_\ell^k \notin [\{\vee, \wedge, \oplus\}]$. Therefore $f_\ell^k(0, \ldots, 0) = 1$. Since the constant unary Boolean function c_0 is in R_0, the function c_0^k that maps all words from $\{0,1\}^+$ to 0^k is in FCPS_B. So the constant function c_1 with $c_1(x) = 1$, for all $x \in \mathbb{N}$, is in FCPS_B, since $c_1(x) = \mathrm{id}_\ell(f(c_0^k(x)))$.

Now let $g \in \mathrm{FP}$. Then there is a uniform family of circuits $(G_n)_{n\in\mathbb{N}}$ over $\{\wedge, \neg\}$ that computes g. Let $g' \in \mathrm{FCPS}_B$ be computed by the uniform family of circuits $(E_n)_{n\in\mathbb{N}}$ that is derived from $(G_n)_{n\in\mathbb{N}}$ as follows. Let E_1 map the input

constantly to 1. Construct E_{n+1} from G_n by introducing a new input gate x_0 and replacing every $\neg$ gate with input y by an $\oplus$ gate with inputs x_0 and y. Obviously, for all $x \in \{0,1\}^+$, $g'(1x) = g(x)$. Therefore $g(x) = g'(\mathrm{conc}(c_1(x), x))$ is in FCPS_B.
Again, the proof for $[B] = \mathrm{R}_1$ is as above, but $\wedge$ is switched with $\vee$, $\oplus$ is switched with $\leftrightarrow$, and 0 is switched with 1.

As mentioned before, FCPS_B is not precomplete for FP if $[B] = \mathrm{M}$, because we can efficiently describe monotonic functions with circuits over $\{\mathrm{and}, \mathrm{or}, \mathrm{not}\}$, that cannot be efficiently described by circuits over $\{\mathrm{and}, \mathrm{or}\}$ [Raz85].

Theorem 5. *If* $[B] = \mathrm{M}$*, the clone* FCPS_B *is not precomplete for* FP.

Proof. Razborov [Raz85] proved a superpolynomial lower bound for the size of monotone circuits computing the perfect matching function. Since this function is in FP, it can be computed by a family of circuits of polynomial size over $\{\wedge, \vee, \neg\}$. That means that there is a function f in $\mathrm{FP} \setminus \mathrm{FCPS}_B$ that can be expressed by a family of monotone circuits. Therefore $[\mathrm{FCPS}_B \cup \{f\}]$ contains just functions computable by monotonic circuits, but not, for example, the function that flips just the first bit of the input, which is a nonmonotonic function that is clearly in FP.

There remains just the one case were $[B] = \mathrm{L}$. In this case FCPS_B, is no dualatom in $\mathcal{L}(\mathrm{FP})$, too: Per definition, every linear Boolean function can be described by a very short propositional formula, and therefore every $f \in \mathrm{FCPS}_B$ can be described by circuits of very small depth. Hence we can use Smolensky's theorem [Smo87] to show that FCPS_B is not a dualatom in $\mathcal{L}(\mathrm{FP})$. We need a few definitions for that.

Definition 6. *Let* $y \in \mathbb{N}$ *such that* $\mathrm{bin}(y) = a_1 \ldots a_m$*,* $n > 0$ *and* $x_1, \ldots, x_n \in \{0,1\}$*. We define*
- $\mathrm{mod}_p^n(x_1, \ldots, x_n) =_{\mathrm{df}} 1$ *if and only if* $\Sigma_{i=1}^n x_i \equiv 0 (\mathrm{mod}\ p)$
- $\mathrm{mod}_p(y) =_{\mathrm{df}} \mathrm{mod}_p^m(a_1, \ldots, a_m)$.
- $\vee^n(x_1, \ldots, x_n) =_{\mathrm{df}} 1$ *if and only if* $\Sigma_{i=1}^n x_i > 0$
- $\wedge^n(x_1, \ldots, x_n) =_{\mathrm{df}} 1$ *if and only if* $\Sigma_{i=1}^n x_i = n$
- $\mathrm{MOD}_r =_{\mathrm{df}} \{\neg\} \cup \bigcup_{n \in \mathbb{N}} \{\mathrm{mod}_p^n, \vee^n, \wedge^n\}$

Theorem 6 (Smolensky). *Let* p *be prime and* $r > 1$ *be relatively prime to* p*. Then* $\mathrm{mod}_r \notin \text{FSIZE-DEPTH}_{\mathrm{MOD}_p}(O(n^{O(1)}), O(1))$.

Corollary 2. *If* $[B] = \mathrm{L}$ *then* FCPS_B *is not precomplete for* FP.

Proof. By definition, $\mathrm{FCPS}_B \subseteq \text{FSIZE-DEPTH}_{\mathrm{MOD}_2}(n^{O(1)}, O(1)) \subset \mathrm{FP}$, since by Theorem 6 $\mathrm{mod}_p \notin \text{FSIZE-DEPTH}_{\mathrm{MOD}_2}(n^{O(1)}, O(1))$ for all primes $p \neq 2$.

Corollary 3. FCPS_B *is a dualatom in* $\mathcal{L}(\mathrm{FP})$ *if and only if* $[B] \in \{\mathrm{R}_0, \mathrm{R}_1, \mathrm{D}\}$.

The dualatoms we found so far contain functions that are nearly as powerful as the most complex ones of FP are. We made slight syntactical restrictions on the functions in FP or on the computation model for these to define precomplete classes. It would be very interesting to know whether there are precomplete classes for

FP that are weaker in a stricter sense. That means whether there are precomplete classes A such that we cannot easily encode every problem from P such that it can be decided by functions from A. One way to do this could be to reduce the resources of a Turing machine. In the following, we look at subclones of FP that are defined via Turing machines with a time bound that is less than polynomial. In particular, we show that there is no $\mathcal{K}$ such that $[\mathrm{DTIME}(\mathcal{K})]$ is precomplete for FP. The technique we use is similar to the one used to show that there exist infinitely many turing degrees between P and NP if P $\neq$ NP [Lad75,Sch82]. We define classes of functions that on large areas of $\mathbb{N}$ produce very short, subpolynomial outputs and on other areas produce outputs of polynomial size. We find such classes that are in fact clones and show that every $[\mathrm{DTIME}(\mathcal{K})]$ is either contained in one of these or already contains FP.
For a set $B \subseteq \mathbb{N}^{\mathbb{N}}$ of unary functions, let

$$\begin{aligned}[B]_{+\mathrm{fict}} =_{\mathrm{df}} \{\ & f : f \in \mathbb{N}^n \text{ for some } n \geq 1 \text{ and } f(x_1,\ldots,x_n) = f'(x_i) \\ & \text{for an } f' \in B \text{ and } 1 \leq i \leq n \text{ and all } x_1,\ldots,x_n \in \mathbb{N}\ \}.\end{aligned}$$

That means, that $[B]_{+\mathrm{fict}}$ contains, all functions f that have at most one nonfictive variable and on this variable f behaves like a function from B. Clearly, $B \subset [B]_{+\mathrm{fict}}$ and if B is closed under substitution, $[B]_{+\mathrm{fict}}$ is closed under superposition. We use this construction in order to not have to deal with all the operations of superposition.

Definition 7. *For $k \in \mathbb{N}$ and $\varepsilon > 0$ say $g : \mathbb{N} \to \mathbb{N}$ is a (k,ε)-*gap function *if and only if for all $\ell \geq k$ holds $g(\ell+1) > 2^{a^{b^c}} - 1$ where $a =_{\mathrm{df}} |g(\ell)| - 1$, $b =_{\mathrm{df}} 1+\varepsilon$, and $c =_{\mathrm{df}} \ell - k + 1$. Define $\mathrm{rgb}_g(k,\ell) =_{\mathrm{df}} 2^{a^{b^c}} - 1$ (*__r__*ight* __g__*ap* __b__*ound). Obviously, if g is a (k,ε)-gap function, it is also a (k',ε)-gap function for all $k' \geq k$.*
We say a function f has (g,k)-gaps *if there is an $\varepsilon > 0$ such that g is a (k,ε)-gap function and for all $\varepsilon' > 0$ we have*

$$x \in [g(\ell), \mathrm{rgb}_g(k,\ell)] \Rightarrow |f(x)| \leq |x|^{1+\varepsilon'}.$$

For a (k,ε)-gap function g, let

$$\mathrm{GAP}_g =_{\mathrm{df}} \{f : \text{ there is a } k' \geq k \text{ such that } f \text{ has } (g,k')\text{-gaps}\}.$$

Let $\mathrm{GAPP}_g =_{\mathrm{df}} \mathrm{FP} \cap [\mathrm{GAP}_g]_{+\mathrm{fict}}$.

We first show some properties of gap functions.

Lemma 2. *Let g be a (k,ε)-gap function, let $k' \geq k$, and let f have (g,k')-gaps.*
1. *For all $\ell \geq k$ holds $|\mathrm{rgb}_g(k,\ell)|^{1+\varepsilon} \leq |\mathrm{rgb}_g(k-1,\ell)|$.*
2. *f has $(g,k'+1)$-gaps.*

Proof. 1. Observe for $a =_{\mathrm{df}} |g(\ell)|$ and $b =_{\mathrm{df}} 1+\varepsilon$:

$$\begin{aligned}|\mathrm{rgb}_g(k,\ell)|^b &= |2^{a^{b^{(\ell-k+1)}}} - 1|^b \\ &= (a^{b^{(\ell-k+1)}})^b \\ &= a^{b^{(\ell-k+2)}} \\ &= |\mathrm{rgb}_g(k-1,\ell)|\end{aligned}$$

2. Obvious, since $\mathrm{rgb}_g(k'+1,\ell) < \mathrm{rgb}_g(k',\ell)$ for all $\ell \geq k'+1$ holds.

The next lemma shows that our classes of gap functions are closed under substitution. We have already seen that this is sufficient for $[\mathrm{GAP}_g]_{+\mathrm{fict}}$ to be a clone.

Lemma 3. *Let g be a (k,ε)-gap function. Then GAP_g is closed under substitution.*

Proof. Let $f_1 \in \mathrm{GAP}_g$ have (g,k_1)-gaps and let $f_2 \in \mathrm{GAP}_g$ have (g,k_2)-gaps. Without loss of generality, let $k_2 \leq k_1$. We show, that $f_1 \circ f_2$ has (g,k_1+1)-gaps. Because of Lemma 2.2 f_2 has (g,k_1+1)-gaps. Therefore, for all $\ell \geq k_1+1$ and all $x \in [g(\ell), \mathrm{rgb}_g(k_1+1,\ell)]$ holds $|f_2(x)| \leq |x|^{1+\varepsilon_1}$ for all $\varepsilon_1 > 0$. Because of Lemma 2.1, for these x we have

$$\begin{aligned}|f_2(x)| &\leq |\mathrm{rgb}_g(k_1+1,\ell)|^{1+\varepsilon_1}\\ &\leq |\mathrm{rgb}_g(k_1,\ell)|\end{aligned}$$

for all $\varepsilon_1 > 0$. Therefore, since f_1 has (g,k_1)-gaps,

$$\begin{aligned}|f_1(f_2(x))| &\leq (|x|^{1+\varepsilon_1})^{1+\varepsilon_2}\\ &= |x|^{1+\varepsilon_1+\varepsilon_2+\varepsilon_1\varepsilon_2}\end{aligned}$$

for all $\varepsilon_1, \varepsilon_2 > 0$.

Corollary 4. *For all (k,ε)-gap functions g, the class GAPP_g is a proper subclone of* FP.

Proof. Since GAPP_g is closed under superposition and $\mathrm{GAPP}_g \subseteq \mathrm{FP}$, it is a subclone of FP. Since the function $s(x) =_{\mathrm{df}} 1^{|x|^2} \in \mathrm{FP}$ has no (g,k)-gaps, for any gap function g, the inclusion is proper.

We have already seen in Proposition 2 that for every class of functions $\mathcal{K}$ that contains a function t with $t(n) \geq n^{1+\varepsilon}$, for some $\varepsilon > 0$, we have $\mathrm{FP} \subseteq [\mathrm{DTIME}(\mathcal{K})]$. We now want to study those $[\mathrm{DTIME}(\mathcal{K})]$ where every function in $\mathcal{K}$ is smaller than $n^{1+\varepsilon}$ for all $\varepsilon > 0$. It is obvious, that all these classes are contained in the class SUBP we define below. SUBP stands for **subp**olynomial time.

Definition 8. *Let* $\mathrm{SUBP} =_{\mathrm{df}} \bigcap_{\varepsilon>0} \mathrm{DTIME}(O(n^{1+\varepsilon}))$.

Theorem 7. SUBP *is a subclone of* FP *and it is no dualatom in the clone lattice below* FP.

Proof. Obviously, $\mathrm{SUBP} \subseteq \mathrm{FP}$. We show that it is a clone. For that let $f_1, f_2 \in \mathrm{SUBP}$. Then there is a $k \in \mathbb{N}$ such that for all $x \in \mathbb{N}$ and all $\varepsilon > 0$ holds $|f_2(x)| \leq k|x|^{1+\varepsilon}$. So, $f_1(f_2(x))$ can be computed in less than $c(k|x|^{1+\varepsilon})^{1+\varepsilon} = k'|x|^{1+2\varepsilon+\varepsilon^2}$ time, for some constants $k', c \in \mathbb{N}$ and all $\varepsilon > 0$.
To show that SUBP is not a dualatom, we show that $\mathrm{SUBP} \subset \mathrm{GAPP}_g$ for a (k,ε)-gap function g. For that, let $\varepsilon =_{\mathrm{df}} 1$, $k = 2$, and $g(x) =_{\mathrm{df}} \exp(4, 2x)$. Here $\exp(0,x) =_{\mathrm{df}} x$ and $\exp(i+1,x) =_{\mathrm{df}} 2^{\exp(i,x)}$ for all $i \geq 0$. Observe, that g is indeed a $(2,1)$-gap function. Let $s(x) =_{\mathrm{df}} \exp(1, (|x|-1)^2)$ and

$$f(x) =_{\mathrm{df}} \begin{cases} s(x) & \text{, if } x = \exp(4, 2\ell+1) \text{ for an } \ell \geq 1\\ 0 & \text{otherwise}\end{cases}$$

Obviously, $f \notin \mathrm{SUBP}$. We show that $f \in \mathrm{GAPP}_g$. For that, let $\ell \geq 2$. Observe that for $c = \ell - k + 1$ we have

$$\begin{aligned}
\mathrm{rgb}_g(2, \ell) &= 2^{(|g(\ell)|-1)^{2^c}} - 1 \\
&< 2^{\exp(3,2\ell)^{2^c}} \\
&= \exp(2, 2^c \cdot \exp(2, 2\ell)) \\
&= \exp(3, \ell - 1 + 2^{2\ell}) \\
&< \exp(4, 2\ell + 1)
\end{aligned}$$

So for all $x \in [g(\ell), \mathrm{rgb}_g(2, \ell)]$ holds $f(x) = 0$ and since $f \in \mathrm{FP}$, it is also in GAPP_g. Therefore $\mathrm{SUBP} \subset \mathrm{GAPP}_g$ and $[\mathrm{SUBP} \cup \{f\}] \subseteq \mathrm{GAPP}_g \subset \mathrm{FP}$.

Corollary 5. *For all classes $\mathcal{K}$ of functions,* $[\mathrm{DTIME}(\mathcal{K})]$ *is not a dualatom in the lattice of clones below* FP.

Proof. Suppose there is an $\varepsilon > 0$ such that for a function $f \in \mathrm{DTIME}(\mathcal{K})$ holds $f(n) \geq n^{1+\varepsilon}$ for all but finite n. Then $\mathrm{FP} \subseteq [\mathrm{DTIME}(\mathcal{K})]$ because of Proposition 2.
On the other hand, if for all $\varepsilon > 0$ and every $f \in \mathrm{DTIME}(\mathcal{K})$ we have $f(n) < n^{1+\varepsilon}$, then $[\mathrm{DTIME}(\mathcal{K})] \subseteq \mathrm{SUBP} \subset \mathrm{GAPP}_g \subset \mathrm{FP}$ for a proper (k, ε)-gap function g.

We remark, that the GAPP_g clones are not precomplete for FP, since for a gap function g we can always construct a gap function $g'(\ell) = g(2\ell)$. Then $\mathrm{GAPP}_g \subset \mathrm{GAPP}_{g'} \subset \mathrm{FP}$.

3.3 The Size of the Lattice

The number of functions in FP is countable. This means that the "depth" of $\mathcal{L}(\mathrm{FP})$, i.e. the number of clones on a path from the lowest element of $\mathcal{L}(\mathrm{FP})$ to FP is countable: For all $A, B \in \mathcal{L}(\mathrm{FP})$ with $A \subset B$ there is a $f \in B \setminus A$. Since $\mathcal{L}(\mathrm{BF})$ is a sublattice of $\mathcal{L}(\mathrm{FP})$, and since the depth of $\mathcal{L}(\mathrm{BF})$ is infinite, so is that of $\mathcal{L}(\mathrm{FP})$. Is the number of clones in $\mathcal{L}(\mathrm{FP})$ also countable? We show that there are uncountably many clones in $\mathcal{L}(\mathrm{FP})$, which implies that most of them have only infinite bases.
For that we introduce *transposition functions*. The idea for these stems from the self-dual Boolean functions. We replace the role of the Boolean not in their definition (see Definition 2) by transposition functions.

Definition 9. *We call a function $g : \mathbb{N} \to \mathbb{N}$* transposition function *if for all $x \in \mathbb{N}$ holds $g(g(x)) = x$. For each such g we define the class of g-*transposition dual functions

$$\mathrm{TD}_g =_{\mathrm{df}} \{f : \forall x_1, \ldots x_k \in \mathbb{N}(f(x_1, \ldots, x_k) = g(f(g(x_1), \ldots, g(x_k))))\}.$$

Let $\mathrm{TDP}_g =_{\mathrm{df}} \mathrm{TD}_g \cap \mathrm{FP}$.

So unary f are in TD_g if and only if $f \circ g = g \circ f$, i.e. f and g commute. For a transposition function g, let $\mathrm{NF}_g =_{\mathrm{df}} \{x : g(x) \neq x\}$ be the set of numbers that are **n**o **f**ixpoints of g.

Proposition 4. *For each transposition function g is TD_g a clone.*

Proof. Every TD_g is obviously closed under ZV, LV, and ID. Let $f_1, f_2 \in \mathrm{TD}_g$, f_1 be m-ary and f_2 be n-ary, and let

$$h(x_1, \dots, x_{m-1}, y_1, \dots, y_n) =_{\mathrm{df}} f_1(x_1, \dots, x_{m-1}, f_2(y_1, \dots, y_n)).$$

Then the following equation holds:

$$\begin{aligned} g(h(g(x_1), \dots, g(y_n))) &= g(f_1(g(x_1), \dots, g(x_{m-1}), f_2(g(y_1), \dots, g(y_m)))) \\ &= g(f_1(g(x_1), \dots, g(x_{m-1}), g(g(f_2(g(y_1), \dots, g(y_m)))))) \\ &= g(f_1(g(x_1), \dots, g(x_{m-1}), g(f_2(y_1, \dots, y_m)))) \\ &= f_1(x_1, \dots, x_{m-1}, f_2(y_1, \dots, y_m)) \\ &= h(x_1, \dots, y_n) \end{aligned}$$

Proposition 5. *Let g_1, g_2 are transposition functions with $g_1 \neq g_2$ and $g_1, g_2 \notin \{\mathrm{id}\}$.*
1. *Then $\mathrm{TD}_{g_1} \not\subseteq \mathrm{TD}_{g_2}$ and $\mathrm{TD}_{g_2} \not\subseteq \mathrm{TD}_{g_1}$.*
2. *If not both $\mathrm{NF}_{g_1} = \mathrm{NF}_{g_2}$ and $|\mathrm{NF}_{g_1}| = |\mathbb{N}|$ then $\mathrm{TDP}_{g_1} \not\subseteq \mathrm{TDP}_{g_2}$ and $\mathrm{TDP}_{g_2} \not\subseteq \mathrm{TDP}_{g_1}$.*

Proof. **Case 1:** $\mathrm{NF}_{g_1} = \mathrm{NF}_{g_2} =_{\mathrm{df}} \{a_0, a_1, \dots\}$. First, let NF_{g_1} be infinite. Let $c_0 =_{\mathrm{df}} a_0$ and $d_0 =_{\mathrm{df}} g_1(a_0)$. For $i > 0$, let c_i be the smallest $a \in \mathrm{NF}_{g_1}$ such that $a \neq c_j$ and $a \neq d_j$ for all $j < i$ and let $d_i = g_1(c_i)$. We define

$$f(x) =_{\mathrm{df}} \begin{cases} c_{i+1} & \text{, if } x = c_i \text{ for an } i \in \mathbb{N} \\ d_{i+1} & \text{, if } x = d_i \text{ for an } i \in \mathbb{N} \\ x & \text{otherwise} \end{cases}$$

Then for an $i \in \mathbb{N}$ and $x \notin \mathrm{NF}_{g_1}$ holds

$$\begin{aligned} g_1(f(g_1(c_i))) &= g_1(f(d_i)) = g_1(d_{i+1}) = c_{i+1} = f(c_i) \\ g_1(f(g_1(d_i))) &= g_1(f(c_i)) = g_1(c_{i+1}) = d_{i+1} = f(d_i) \\ g_1(f(g_1(x))) &= g_1(f(x)) = g_1(x) = x = f(x). \end{aligned}$$

So $f \in \mathrm{TD}_{g_1}$. Let us assum that $f \in \mathrm{TD}_{g_2}$. Since $g_1 \neq g_2$ there is a smallest i such that $g_1(c_i) \neq g_2(c_i)$. Hence, there is a $j > 0$ such that $g_2(c_i) = c_{i+j}$ or $g_2(c_i) = d_{i+j}$.
Let $g_2(c_i) = c_{i+j}$. Then for all $k \geq 0$ holds $g_2(c_{i+k}) = c_{i+j+k}$ since $c_{i+k+1} = f(c_{i+k}) = g_2(f(g_2(c_{i+k}))) = g_2(f(c_{i+j+k})) = g_2(c_{i+j+k+1})$. For $k = j$ this leads to $c_{i+2j} = g_2(c_{i+j}) = c_i$ which is a contradiction.
Now let $g_2(c_i) = d_{i+j}$. Then for all $k \geq 0$ holds $g_2(c_{i+k}) = d_{i+j+k}$ since $d_{i+j+k+1} = f(d_{i+j+k}) = g_2(f(g_2(d_{i+j+k}))) = g_2(f(c_{i+k})) = g_2(c_{i+k+1})$. So for all $i \in \mathbb{N}$ we have determined the value of $g_2(c_i)$. Furthermore, we know the value of all $g_2(d_k)$ if $k < i$ or $k \geq i + j$. Therefore, there is a $0 < \ell < j$ such that $g_2(d_i) = d_{i+\ell}$ and with the same argumentation as above, we obtain for all $k \geq 0$ the equation $g_2(d_{i+k}) = d_{i+\ell+k}$. For $k = j$ that means $c_i = g_2(d_{i+j}) = d_{i+j+\ell}$ which is a contradiction.
If NF_{g_1} is finite, then there is a $y \in \mathbb{N} \setminus \mathrm{NF}_{g_1}$. Let $a, b \in \mathrm{NF}_{g_1}$ such that $g_1(a) = b$ and $g_2(a) = c \neq b$ and define

$$f(x) =_{\mathrm{df}} \begin{cases} y & \text{, if } x \in \{a, b\} \\ x & \text{otherwise} \end{cases}$$

Then $g_1(f(g_1(a))) = g_1(f(b)) = g_1(y) = y = f(a)$ and $g_1(f(g_1(x))) = g_1(g_1(x)) = x = f(x)$ for $x \notin \{a,b\}$, so $f \in \mathrm{TD}_{g_1}$. But $g_2(f(g_2(a))) = g_2(f(c)) = g_2(c) \neq y = f(a)$, since $g_2(c) \in \mathrm{NF}_{g_1}$.
Case 2: $\mathrm{NF}_{g_1} \subset \mathrm{NF}_{g_2}$ or ($\mathrm{NF}_{g_2} \not\subseteq \mathrm{NF}_{g_1}$ and $\mathrm{NF}_{g_1} \not\subseteq \mathrm{NF}_{g_2}$). Then there exist $a, b \in \mathrm{NF}_{g_2} \setminus \mathrm{NF}_{g_1}$ such that $g_2(a) = b$ and $g_1(a) = a$. Define $f(x) =_{\mathrm{df}} a$ for all $x \in \mathbb{N}$. Then $g_1(f(g_1(x))) = g_1(a) = a = f(x)$ and therefore $f \in \mathrm{TD}_{g_1}$. But $g_2(f(g_2(x))) = g_2(a) = b \neq a = f(x)$. Hence, $\mathrm{TD}_{g_1} \not\subseteq \mathrm{TD}_{g_2}$.
It remains to show that $\mathrm{TD}_{g_2} \not\subseteq \mathrm{TD}_{g_1}$. For that, let $a' \in \mathrm{NF}_{g_1}$ and define

$$f(x) =_{\mathrm{df}} \begin{cases} a' & \text{, if } x = a \\ g_2(a') & \text{, if } x = b \\ x & \text{otherwise} \end{cases}$$

Then $g_2(f(g_2(a))) = g_2(f(b)) = g_2(g_2(a')) = a' = f(a)$ and $g_2(f(g_2(b))) = g_2(f(a)) = g_2(a') = f(b)$ holds, and if $x \notin \{a,b\}$ then $g_2(f(g_2(x))) = g_2(g_2(x)) = x = f(x)$, since $g_2(x) \notin \{a,b\}$. Therefore $f \in \mathrm{TD}_{g_2}$. Furthermore, $g_1(f(g_1(a))) = g_1(f(a)) = g_1(a') \neq a' = f(a)$, and therefore $f \notin \mathrm{TD}_{g_1}$.
For the third claim, note that all functions f we created above are in FP, if not $\mathrm{NF}_{g_1} = \mathrm{NF}_{g_2}$ and NF_{g_1} infinite.

Corollary 6. *Let $g_1, g_2 \notin \{\mathrm{id}\}$ be transposition functions with $g_1 \neq g_2$.*
1. *There is no transposition function g such that $\mathrm{TD}g_1 \cap \mathrm{TD}g_2 = \mathrm{TD}_g$.*
2. *There is no transposition function $g \neq \mathrm{id}$ such that $[\mathrm{TD}g_1 \cup \mathrm{TD}g_2] = \mathrm{TD}_g$.*

Corollary 7. *There are uncountably many clones in $\mathcal{L}(\mathrm{FP})$.*

Acknowledgements: I would like to thank Christian Glaßer, Daniel Meister, Steffen Reith, and Heribert Vollmer for helpful hints and discussions.

References

[BCRV03] E. Böhler, N. Creignou, S. Reith, and H. Vollmer. Playing with Boolean blocks, part I: Post's lattice with applications to complexity theory. *ACM-SIGACT Newsletter*, 34(4):38–52, 2003.

[Clo99] P. Clote. Computation models and function algebras. *Handbook of Computability Theory*, pages 589–681, 1999.

[Coh65] P. M. Cohn. *Universal Algebra*. Harper & Row, New York, Evanston, London and John Wheatherhill, Inc., Tokyo, first edition, 1965.

[Grä79] G. Grätzer. *Universal Algebra*. Springer Verlag, Berlin Heidelberg New York, second edition, 1979.

[JGK70] S. W. Jablonski, G. P. Gawrilow, and W. B. Kudrajawzew. *Boolesche Funktionen und Postsche Klassen*. Akademie-Verlag, 1970.

[Lad75] R. Ladner. On the structure of polynomial-time reducibility. *Journal of the ACM*, 22:155–171, 1975.

[Meh76] Kurt Mehlhorn. Polynomial and abstract subrecursive classes. *J. Comput. Syst. Sci.*, 12(2):147–178, 1976.

[Neu37] B. H. Neumann. Some remarks on inifinite groups. *J. London Math. Soc.*, 12:120–127, 1937.

[Pos41] E. L. Post. The two-valued iterative systems of mathematical logic. *Annals of Mathematics Studies*, 5, 1941.

[Raz85] A. A. Razborov. Lower bounds on the monotone complexity of some boolean functions. *Dokl. Akad. Nauk*, 281:798–801, 1985.

[Sch82] U. Schöning. A uniform approach to obtain diagonal sets in complexity classes. *Theoretical Computer Science*, 18:95–103, 1982.

[Smo87] R. Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proceedings 19th Symposium on Theory of Computations*, pages 77–82. ACM Press, 1987.

[Vol99] H. Vollmer. *Introduction to Circuit Complexity*. Springer, Berlin Heidelberg New York, 1999.

[VW96] H. Vollmer and K. Wagner. Recursion theoretic characterizations of complexity classes of counting functions. *Theoretical Computer Science*, 163:245–258, 1996.

[Wag86] K. Wagner. Bounded recursion and complexity classes. *Theoretical Computer Science*, 47:131–147, 1986.