Electronic Colloquium on Computational Complexity, Report No. 77 (2000)

On the Power of Extra Queries to Selective Languages

Till Tantau
Technische Universität Berlin
tantau@cs.tu-berlin.de

24th August 2000

Abstract

A language is *selective* if there exists a selection algorithm for it. Such an algorithm selects from any two words one, which is an element of the language whenever at least one of them is. Restricting the complexity of selection algorithms yields different *selectivity classes* like the P-selective or the semirecursive (i. e. recursively selective) languages. A language is *supportive* if k queries to the language are more powerful than $k-1$ queries for every k. Recently, Beigel et al. [4] proved a powerful recursion theoretic theorem: A semirecursive language is supportive iff it is nonrecursive. For restricted computational models like polynomial time this theorem does not hold in this form. Our main result states that for any reasonable computational model *a selective language is supportive iff it is not cheatable.* Beigel et al.'s result is a corollary of this general theorem since 'recursively cheatable' languages are recursive by Beigel's Nonspeedup Theorem [2]. Our proof is based on a partial information analysis [17, 18] of the involved languages: We establish matching upper and lower bounds for the partial information complexity of the equivalence and reduction closures of selective languages. From this we derive the main results as these bounds differ for different k.

We give four applications of our main theorem and the proof technique. Firstly, the relation $\mathrm{E}^{\mathrm{P}}_{k\text{-tt}}(\mathrm{P}[\mathrm{SEL}]) \not\subseteq \mathrm{R}^{\mathrm{P}}_{(k-1)\text{-tt}}(\mathrm{P}[\mathrm{SEL}])$ proven in [12] still holds, *if we relativise only the right hand side.* Secondly, we settle an open problem from [12]: Equivalence to a P-selective language with k *serial* queries cannot generally be replaced by a reduction using less than 2^k-1 *parallel* queries. Thirdly, the k-truth-table reduction closures of selectivity classes are (m,n)-verbose [7] iff every walk on the n-dimensional hypercube with transition counts at most k visits at most m bitstrings. Lastly, these reduction closures are (m,n)-recursive [21] iff every such walk is contained in a closed ball of radius $n-m$.

Keywords

Structural complexity, bounded query complexity, P-selective, semirecursive, supportive, cheatable, verbose, frequency computations.

A *selector* for a language $L \subseteq \Sigma^*$ is a binary function $f\colon \Sigma^* \times \Sigma^* \to \Sigma^*$ such that $f(u,v) \in \{u,v\}$, and $f(u,v) \in L$ whenever $u \in L$ or $v \in L$. Jockusch [13] coined the term *semirecursive languages* for languages which have a recursive selector. The class of semirecursive languages, denoted by $\mathrm{REC}[\mathrm{SEL}]$ in the following, plays a key role in the solution of Post's Problem [19].

ISSN 1433-8092

Selman [22] introduced P-*selective languages*. Such languages have a polynomial time computable selector. The class of P-selective languages will be denoted by P[SEL], see Definitions 9 and 14 for the reasons for this notation. Selman proved that the satisfiability problem is not P-selective, unless P = NP. This result has been considerably strengthened by extending it to *reduction closures* of the P-selective languages, see the following Facts 2 and 3.

Fact 1 ([22]). *If some* NP*-hard language is* P*-selective, then* NP = P.

Fact 2 ([6, 20, 1]). *If some* NP*-hard language is sublinear truth-table reducible to a* P*-selective language, then* NP = P.

Fact 3 ([22, 15]). *If some* NP*-hard language is Turing reducible to a* P*-selective language, then* $\text{NP} \subseteq \text{P/poly}$.

These three facts have been a major motivation for the study of the power of extra queries to P-selective languages. Hemaspaandra et al. [12] proved that each extra query to the class of P-selective languages helps to decide harder problems, see Fact 4. Following [4], we call a class of languages *supportive* if its bounded Turing reduction closures form a proper hierarchy. We call it *parallel supportive* if the same is true for the truth-table closures.

Fact 4 ([12]). *The class* P[SEL] *is both polynomial time supportive and parallel polynomial time supportive.*

Recently, using recursion theoretic methods Beigel et al. proved the much more powerful Fact 5. The complexity theoretic Fact 4 follows from an extended version of the recursion theoretic Fact 5, see Corollary 30.

Fact 5 ([4]). *A semirecursive language is supportive and parallel supportive iff it is nonrecursive.*

Just like Facts 4 and 5, our results hold for both serial and parallel queries. However, due to lack of space we formulate theorems only for *parallel* queries in the following. The transferal to serial queries is trivial for most computational models, including polynomial time computations – see Fact 21 for details.

Hemaspaandra et al.'s proof of Fact 4 only shows that the *class* P[SEL] is parallel polynomial time supportive, but not that any individual language in it is. Indeed, the languages constructed in their diagonalisation are *cheatable*, see Definition 10, and thus not parallel supportive. This shows that Fact 5 cannot be directly transferred to polynomial time, since there are P-selective languages outside P that are not parallel polynomial time supportive.

Our main result, Theorem 28, is a generalisation of Fact 5 which holds for both polynomial time *and* the recursive case. It states that for any reasonable computational model C, *a C-selective language is parallel C-supportive iff it is not C-cheatable.* As 'recursively cheatable' languages are recursive by Beigel's Nonspeedup Theorem [2], Fact 5 becomes a corollary.

This paper is organised as follows. *Section 1* studies logspace selective languages and intends to motivate why the study of selectivity should not be restricted to P-selectivity and semirecursiveness. We transfer Fact 1 to P-hard and NL-hard languages, see Corollaries 7 and 8, but fail to prove transferred versions of Facts 2 and 3, even if the reduction is restricted to a single query.

Section 2 introduces basic notations and definitions. It includes a review of the main technical tool used in our proofs: partial information classes, a concept proposed in [5] and studied extensively by Nickelsen [17, 18]. In *Section 3* we establish matching upper and lower bounds on the partial information complexity of languages reducible to a partial information class. *Section 4* shows that for the special case of selectivity classes these bounds are walks on a multi-dimensional hypercube with bounded transition counts. We deduce that selective languages are parallel supportive iff they are not cheatable, by noting that for different transition counts these walks visit different numbers of bitstrings.

The remaining sections present applications of the main result and its proof technique. In *Section 5* we prove that for every oracle A we have $\mathrm{E}^{\mathrm{P}}_{k\text{-tt}}(\mathrm{P}[\mathrm{SEL}]) \not\subseteq \mathrm{R}^{\mathrm{P}^A}_{(k-1)\text{-tt}}(\mathrm{P}^A[\mathrm{SEL}])$. This is a powerful relativisation of the corresponding unrelativised relation proven in [12]. In *Section 6* we settle an open problem from [12]: There is a language which is k-Turing equivalent to a P-selective language, but not $(2^k - 2)$-truth-table reducible to any P-selective language.

Section 7 shows how our proof technique can be used to quantify the amount of *verboseness* of reduction closures of selective languages; *Section 8* does the same for frequency computations. The verboseness of reduction closures of selective languages has been studied implicitly in [7]. We rephrase, in terms of walks and transition counts, the combinatorial characterisation established in [7]. This rephrasing shows that a conjecture from [7] seems hard to prove since it implies the existence of balanced Gray codes for arbitrary bitstring lengths.

1 Selectivity of P- and NL-Complete Problems

Selman [22] showed that the satisfiability problem is not P-selective, unless P = NP. The proof exploits the self-reducibility of SAT and shows how a selector can be used to decide the satisfiability of a formula in polynomial time. Two corollaries below show that the P-complete *circuit value problem* and the NL-complete *directed reachability problem* are not logspace selective, L-selective in short, unless P = L and NL = L respectively.

Theorem 6. *Let X be a class of languages closed under complement. If there exists a language which is logspace many-one complete for X and* L*-selective, then $X \subseteq \mathrm{L}$.*

Proof. Let N be a such an L-selective language. It suffices to show $N \in \mathrm{L}$. As X is closed under complement and N is complete for X, we have $\bar{N} \in X$ and hence $\bar{N} \leq^{\mathrm{L}}_{\mathrm{m}} N$ via some logspace computable reduction R. For an input u compute $v := R(u)$. Note that $v \in N$ iff $u \notin N$. As exactly one of the words u and v is in N, applying the postulated selector to them tells us which one is. □

As P and NL are closed under complement we get two corollaries:

Corollary 7. *If the circuit value problem is* L*-selective, then* L = P.

Corollary 8. *If the directed reachability problem is* L*-selective, then* L = NL.

We do not know whether strengthened versions of the above corollaries hold, but conjecture that *if the circuit value problem is logspace truth-table reducible to an* L*-selective language with a single query, then* L = P.

2 Preliminaries

We start this section with basic notations and definitions. Next, we review the concept of partial information as introduced in [17, 18]. Finally, we state some basic facts about truth-table reductions.

Basic Notations and Definitions. For any n, $\langle \cdot, \dots, \cdot \rangle : (\Sigma^*)^n \to \Sigma^*$ is a tupling function computable in logspace. For a language $L \subseteq \Sigma^*$ and a word $w \in \Sigma^*$ the *characteristic value* $\chi_L(w)$ is defined by $\chi_L(w) := 1$ if $w \in L$, and $\chi_L(w) := 0$ otherwise. This is extended to tuples by setting $\chi_L(w_1, \dots, w_n) := \chi_L(w_1) \dots \chi_L(w_n)$. Let $\mathbb{B} := \{0, 1\}$. *Bitstrings of length n* are elements of $\mathbb{B}^n$. The *Cartesian product* $\phi_1 \times \cdots \times \phi_n \colon \mathbb{B}^{nk} \to \mathbb{B}^n$ of functions $\phi_1, \dots, \phi_n \colon \mathbb{B}^k \to \mathbb{B}$ is defined by $(\phi_1 \times \cdots \times \phi_n)(b_1 \dots b_n) := \phi_1(b_1) \dots \phi_n(b_n)$ for $b_i \in \mathbb{B}^k$. For $\phi \colon \mathbb{B}^n \to \mathbb{B}^m$ and a set $P \subseteq \mathbb{B}^n$ the *ϕ-image of P* is $\phi(P) := \{\, \phi(b) \mid b \in P \,\}$. The *Hamming distance* $\mathrm{d}(b, c)$ of two bitstrings is the number of positions where they differ. The *closed ball around b of radius r* is $\mathrm{B}_r(b) := \{\, c \in \mathbb{B}^n \mid \mathrm{d}(b, c) \leq r \,\}$.

A *walk* is a sequence of bitstrings of the same length where consecutive bitstrings differ at exactly one position. A walk is *self-avoiding*, if the sequence contains no duplicates except possibly for the first and last bitstring. If these are equal, the walk is called a *cycle*. The *transition sequence* of a walk is the sequence of position indices where consecutive bitstrings differ. The *transition count* of a position index is its frequency in a transition sequence. An example of a walk is $001, 011, 010, 110, 111$; its transition sequence is $2, 3, 1, 3$; and the transition count of the first two position indices is 1 and for the last position index it is 2.

Partial Information Classes. Traditionally the complexity of a language L is measured by the amount of time or space needed to compute for words $w_1, \dots, w_n$ the characteristic string $\chi_L(w_1, \dots, w_n)$. In a *partial information analysis*, first used in [5] and put onto a firm theoretical foundation in [17, 18], we consider witness functions f such that $\chi_L(w_1, \dots, w_n) \in f(w_1, \dots, w_n)$ for all words. The function f may produce whole sets of bitstrings, called *n-pools* in the following. We now ask, which are the smallest and simplest pools computable in, say, polynomial time for a given language L?

We will say that *Q is a pool for the words $w_1, \dots, w_n$*, if Q contains their characteristic string. For every non-trivial language every bitstring in $\mathbb{B}^n$ is the characteristic string of some appropriate words $w_1, \dots, w_n$. Hence the set of all pools which a witness function outputs must necessarily form a covering of $\mathbb{B}^n$, called *n-covering* in the following. In this paper, we will use the following special n-coverings:

Definition 9. Define SEL_n as the set of all chains in the Boolean algebra $\mathbb{B}^n$. Such chains will be called *selective pools*. Phrased differently, SEL_n contains all pools which can be written as $\{b_1, \dots, b_m\}$ such that from each b_i to b_{i+1} only some 0's are changed into 1's.

Definition 10. Define $m\text{-}\mathrm{SIZE}_n := \{\, P \subseteq \mathbb{B}^n \mid |P| \leq m \,\}$ for sizes $m \geq 1$.

Definition 11. Define $\mathrm{CHEAT}_n := n\text{-}\mathrm{SIZE}_n$.

Definition 12. Define $k\text{-}\mathrm{WALKS}_n$ as the set of all pools $P \subseteq \mathbb{B}^n$ which are contained in a walk with transition counts at most k.

Definition 13. Define $r\text{-}\mathrm{FREQ}_n := \{\, P \mid \exists b \in \mathbb{B}^n : P \subseteq \mathrm{B}_r(b) \,\}$ for radii r.

We define partial information classes for different computational models and C will be a variable for such a model. Its only property we will use and be interested in is its corresponding *function class FC*. As is customary in complexity theory, the functions in this class map words to words. In the following definition functions actually map tuples of words to pools. Since coding and decoding of both tuples of words and pools is trivial, we will not explicitly write down the appropriate conversions.

Definition 14. Let C denote a computational model and FC the corresponding function class. Let $\mathcal{D}$ be an n-covering. Then a language $L \subseteq \Sigma^*$ is in the *partial information class* $C[\mathcal{D}]$ (respectively $C_{\text{dist}}[\mathcal{D}]$), if there exists a witness $f \in FC$ such that for all (distinct) words $w_1, \ldots, w_n \in \Sigma^*$ we have

$$\chi_L(w_1, \ldots, w_n) \in f(w_1, \ldots, w_n) \in \mathcal{D}.$$

The class $\mathrm{P}[\mathrm{SEL}_2]$ is exactly the class of P-selective languages, $\mathrm{L}[\mathrm{SEL}_2]$ contains the L-selective languages, and $\mathrm{REC}[\mathrm{SEL}_2]$ contains the semirecursive languages.

Note that we require witness machines to output complete pools. One can also consider the situation where a witness need only *enumerate* the output pool. We introduce a special notation for this situation: A language is in $\mathrm{RE}[\mathcal{D}]$ if there exists a Turing machine that upon input of any n words *enumerates* a pool from $\mathcal{D}$ for them. For example, $\mathrm{RE}[\mathrm{SEL}_2]$ is the class of *weakly semirecursive* [14] languages and $\mathrm{RE}[m\text{-SIZE}_n]$ is the class of (m,n)*-verbose* [7] languages. If a theorem also holds for $\mathrm{RE}[\mathcal{D}]$, we give a second version of it, like Theorem 28′.

A pool is called *maximal* for some covering, if there exists no proper superset of this pool in the covering.

An n-covering $\mathcal{D}$ is *subset closed* if $Q \subseteq P \in \mathcal{D}$ implies $Q \in \mathcal{D}$. In the following, we formulate theorems only for subset closed coverings. For partial information classes in the sense of Definition 14 this is just a convenience. However, for the classes $\mathrm{RE}[\mathcal{D}]$ it makes a difference whether the enumerating Turing machine is allowed to enumerate proper subsets of maximal pools or not. Note that all of the special coverings we defined above are subset closed.

The arguments used in the following proofs are correct for all 'reasonable' FC. Examples of reasonable function classes are FL, FP, FP/poly, $\mathrm{FNPSV_t}$, the class FREC of recursive functions, and any relativisation of these classes.

Definition 15 ([24]). A computational model C and the corresponding function class FC are *reasonable*, if $\mathrm{FL} \subseteq FC$ and FC is closed under composition and tupling, i.e., if $f, g \in FC$ then $f \circ g \in FC$ and $\langle f, g\rangle \in FC$. Here, $\langle f, g\rangle(w) := \langle f(w), g(w)\rangle$.

The following fact motivates our liberal notation $\mathrm{P}[\mathrm{SEL}]$, as it tells us that the index is not important for selective coverings.

Fact 16 ([18, 22]). *For reasonable C we have $C[\mathrm{SEL}_n] = C[\mathrm{SEL}_2]$ for $n \geq 2$.*

For a reasonable model C, a language will be called *C-cheatable* if it is in $C[\mathrm{CHEAT}] := \bigcup_{n=1}^{\infty} C[\mathrm{CHEAT}_n] = \bigcup_{n=1}^{\infty} C[n\text{-SIZE}_n]$. The following fact explains why neither the term 'r.e. cheatable' nor the term 'recursively cheatable' is used.

Fact 17 (Nonspeedup Theorem [2]). *We have* $\mathrm{RE}[\mathrm{CHEAT}] = \mathrm{REC}$.

Bounded Query Reductions. Following [16], we now define the notion of truth-table reduction for arbitrary computational models C.

Definition 18. Let C denote a computational model and FC the corresponding function class. A language L is *k-truth-table C-reducible* to a language N, written $L \leq^{C}_{k\text{-tt}} N$, if there exist a *generator* $g \in FC$ and an *evaluator* $e \in FC$ such that for all words $w \in \Sigma^*$ we have $\chi_L(w) = e\big(w, \chi_N(q_1), \dots, \chi_N(q_k)\big)$. Here, $\langle q_1, \dots, q_k \rangle = g(w)$ are the queries produced by the generator upon input w. For a class X of languages define $\mathrm{R}^{C}_{k\text{-tt}}(X) := \big\{ L \mid \exists N \in X \colon L \leq^{C}_{k\text{-tt}} N \big\}$ and $\mathrm{E}^{C}_{k\text{-tt}}(X) := \big\{ L \mid \exists N \in X \colon L \leq^{C}_{k\text{-tt}} N, N \leq^{C}_{k\text{-tt}} L \big\}$.

Note, that for $FC = \mathrm{FREC}$ we require the evaluator to converge on all inputs. We will also consider *weak truth-table* reductions, written $L \leq_{k\text{-wtt}} N$. Here, the evaluator must only converge when provided with the correct characteristic string of the queries. As this reduction makes sense only in the recursive setting, we omit the superscript.

Definition 19. A class X of languages is *parallel C-supportive* if for all k we have $\mathrm{R}^{C}_{k\text{-tt}}(X) \subsetneq \mathrm{R}^{C}_{(k+1)\text{-tt}}(X)$.

Fact 20 ([2]). *For reasonable C no C-cheatable language is parallel C-supportive.*

Due to lack of space, this paper treats only *parallel* queries. Fact 21 shows that for polynomial time computations this is no loss of generality. Note, that the fact's proof cannot be transferred to weak truth-table reductions. The new Lemma 22 is another example of equipotent reductions to selective languages.

Fact 21 ([12]). *Let N be P-selective. Then $L \leq^{\mathrm{P}}_{k\text{-T}} N$ iff $L \leq^{\mathrm{P}}_{(2^k-1)\text{-tt}} N$.*

Lemma 22. *Let N be P-selective. Then $L \leq^{\mathrm{P}}_{k\text{-tt}} N$ iff $L \leq^{\mathrm{P}}_{k\text{-parity}} N$.*

Proof. We only need to prove the first direction. Assume that $L \leq^{\mathrm{P}}_{k\text{-tt}} N$ via a generator g and an evaluator e. For the parity reduction, upon input of a word w we compute the queries q_i generated by g. Using the selectivity of N we compute a permutation σ such that $\chi_N\big(q_{\sigma(i)}\big) \leq \chi_N\big(q_{\sigma(i+1)}\big)$. Let b_i be the bitstring where the positions $\sigma(1), \dots, \sigma(i)$ are set to 1 and the other positions to 0. We construct new queries as follows: Let $t_1, \dots, t_k$ be the indices where bitflips occur in the sequence $e(w, b_0), e(w, b_1), \dots, e(w, b_n)$. The new queries are $q_{\sigma(t_1)}, \dots, q_{\sigma(t_k)}$. Note that $k \leq n$. If $e(w, 0^n) = 0$, we accept w iff the parity of the answers is 1. If $e(w, 0^n) = 1$, we accept iff it is 0. □

Interestingly, the P-selective languages *share* the two properties stated in Fact 21 and Lemma 22 with NP-complete sets [11, 3]. As pointed out in [9], Fact 21 also holds for non-constant numbers of queries as long as $k(n) \in O(\log n)$. Likewise, it is easily seen that Lemma 22 also holds for arbitrary $k(n)$.

3 Upper and Lower Bounds

We now prove matching upper and lower bounds for the partial information complexity of languages reducible to a partial information class. A special case are selectivity classes, which will be studied in the next section. There we will use the matching bounds to prove our main result.

To fix notations, in this section C denotes a reasonable computational model, n, k are positive integers, and $\mathcal{E}$ is a subset closed nk-covering. The n-covering $\mathcal{D}$ always denotes the *image of $\mathcal{E}$ under products of Boolean functions*, defined by

$$\mathcal{D} := \{ (\phi_1 \times \cdots \times \phi_n)(Q) \mid Q \in \mathcal{E}, \phi_i \colon \mathbb{B}^k \to \mathbb{B} \} .$$

Theorem 23 ($\mathcal{D}$ is Upper Bound). *We have* $\mathrm{R}^C_{k\text{-tt}}(C[\mathcal{E}]) \subseteq C[\mathcal{D}]$.

Proof. Let $L \leq^C_{k\text{-tt}} N \in C[\mathcal{E}]$. To show $L \in C[\mathcal{D}]$ we must compute partial information from $\mathcal{D}$ for any n given words $w_1, \dots, w_n$. As L is k-truth-table reducible to N, for each word w_i the generator yields queries $q^1_i, \dots, q^k_i$ and the evaluator computes $\phi_i := e(w_i, \cdot)$ with $\phi_i(\chi_N(q^1_i, \dots q^k_i)) = \chi_L(w_i)$. As $N \in C[\mathcal{E}]$ we can compute a pool $Q \in \mathcal{E}$ containing $\chi_N(q^1_1, \dots, q^k_1, \dots, q^1_n, \dots, q^k_n)$.

Let $Q = \{b_1, \dots, b_m\}$. One of these bitstrings, say b_j, is the correct value of $\chi_N(q^1_1, \dots, q^k_n)$. But then $(\phi_1 \times \cdots \times \phi_n)(b_j)$ is the correct value of the characteristic string $\chi_L(w_1, \dots, w_n)$. Hence, the $(\phi_1 \times \cdots \times \phi_n)$-image of Q is a pool for the input words and an element of $\mathcal{D}$ by definition. □

Theorem 23′. *We have* $\mathrm{R}_{k\text{-wtt}}(\mathrm{RE}[\mathcal{E}]) \subseteq \mathrm{RE}[\mathcal{D}]$.

Proof. Same as above, except that pools are enumerated, not computed. □

Under certain conditions the upper bound just established is also a lower bound, i.e., there exists no smaller n-covering $\mathcal{D}' \subsetneq \mathcal{D}$ such that $\mathrm{R}^C_{k\text{-tt}}(C[\mathcal{E}]) \subseteq C[\mathcal{D}']$. The following theorem strengthens this claim in several ways. We show that the lower bound holds for certain individual languages in $C[\mathcal{E}]$. We consider logspace equivalence closures instead of arbitrary reduction closures on the left hand side. Finally, we use the larger class $C_{\mathrm{dist}}[\mathcal{D}']$ on the right hand side.

Theorem 24 ($\mathcal{D}$ is Lower Bound). *Let $N \in C[\mathcal{E}]$, but $N \notin C[\mathcal{E}']$ for any subset closed covering $\mathcal{E}' \subsetneq \mathcal{E}$. Then for every subset closed covering $\mathcal{D}' \subsetneq \mathcal{D}$*

$$\mathrm{E}^{\mathrm{L}}_{k\text{-tt}}(N) \nsubseteq C_{\mathrm{dist}}[\mathcal{D}'] \text{ and}$$
$$\mathrm{E}^{\mathrm{L}}_{\lceil \log(k+1) \rceil\text{-T}}(N) \nsubseteq C_{\mathrm{dist}}[\mathcal{D}'].$$

Proof. We argue by contraposition, starting with the truth-table reduction. Let $\mathcal{D}' \subsetneq \mathcal{D}$ such that $\mathrm{E}^{\mathrm{L}}_{k\text{-tt}}(N) \subseteq C_{\mathrm{dist}}[\mathcal{D}']$; we must show that there exists some $\mathcal{E}' \subsetneq \mathcal{E}$ such that $N \in C[\mathcal{E}']$. Let $P \in \mathcal{D}$ be some maximal pool not in $\mathcal{D}'$. By definition of $\mathcal{D}$ there exists some maximal pool $Q \in \mathcal{E}$ together with functions $\phi_i \colon \mathbb{B}^k \to \mathbb{B}$ such that $(\phi_1 \times \cdots \times \phi_n)(Q) = P$. Define $\mathcal{E}' := \mathcal{E} \setminus \{Q\}$ and note that this is a subset closed covering.

We now show $N \in C[\mathcal{E}']$. Let $N \in C[\mathcal{E}]$ via a witness $f \in FC$. Given any nk words $w^1_1, \dots, w^k_n$ we must compute a pool for these words from $\mathcal{E}$ *other than* Q. First, we compute the pool $T := f(w^1_1, \dots, w^k_n) \in \mathcal{E}$. Next, consider $N' := \{ \langle u_1, \dots, u_k, \phi, t \rangle \mid \phi \colon \mathbb{B}^k \to \mathbb{B}, \phi(\chi_L(u_1, \dots, u_k)) = 1, t \in \Sigma^* \}$. It is easily seen that N and N' are logspace k-truth-table equivalent. We form n *distinct* words $z_i := \langle w^1_i, \dots, w^k_i, \phi_i, 0^i \rangle$. As $N' \in \mathrm{E}^{\mathrm{L}}_{k\text{-tt}}(N) \subseteq C_{\mathrm{dist}}[\mathcal{D}']$ we can now compute a pool $R \in \mathcal{D}'$ such that we have $\chi_{N'}(z_1, \dots, z_n) \in R$. Note that $P \nsubseteq R$ and $R \in \mathcal{D}$.

We define a pool $S := \{ b \in \mathbb{B}^{nk} \mid (\phi_1 \times \cdots \times \phi_n)(b) \in R \}$ which is the *inverse image* of R. As

$$\begin{aligned} \chi_{N'}(z_1, \dots, z_n) &= \phi_1(\chi_N(w^1_1, \dots, w^k_1)) \cdots \phi_n(\chi_N(w^1_n, \dots, w^k_n)) \\ &= (\phi_1 \times \cdots \times \phi_n)(\chi_N(w^1_1, \dots, w^k_n)), \end{aligned}$$

the pool S must contain the bitstring $\chi_N(w_1^1, \dots, w_n^k)$, since R contains the bitstring $\chi_{N'}(z_1, \dots, z_n)$. Thus S is a pool for the original words, and so is $S \cap T$. If $S \cap T \neq Q$, we are done, since we can then output $S \cap T \in \mathcal{E}'$. But $S \cap T = Q$ is impossible, since $Q \subseteq S$ implies $P = (\phi_1 \times \dots \times \phi_n)(Q) \subseteq (\phi_1 \times \dots \times \phi_n)(S) \subseteq R$. To complete the proof for the Turing reduction, simply note that N and N' are also $\lceil \log(k+1) \rceil$-Turing equivalent due to Fact 21. □

Theorem 24′. *Let $N \in \mathrm{RE}[\mathcal{E}]$, but $N \notin \mathrm{RE}[\mathcal{E}']$ for any subset closed covering $\mathcal{E}' \subsetneq \mathcal{E}$. Then for every subset closed covering $\mathcal{D}' \subsetneq \mathcal{D}$ we have*

$$\mathrm{E}^{\mathrm{L}}_{k\text{-tt}}(N) \nsubseteq \mathrm{RE}_{\mathrm{dist}}[\mathcal{D}'] \text{ and}$$
$$\mathrm{E}^{\mathrm{L}}_{\lceil \log(k+1) \rceil\text{-T}}(N) \nsubseteq \mathrm{RE}_{\mathrm{dist}}[\mathcal{D}'].$$

Proof. Same as for Theorem 24, except that pools are enumerated instead of computed. Note, that if R can be enumerated, so can S; and if S and T can be enumerated, so can $S \cap T$. □

4 Non-Cheatable Selective Languages are Supportive

This section applies the upper and lower bounds established in Section 3 to selectivity classes. First, we study some combinatorial properties of *images of selective pools under products of Boolean functions*, see Lemmas 25 and 26. Next, we show that *the condition $N \notin C[\mathcal{E}']$ from Theorem 24 is met* for non-cheatable N, see Lemma 27. Put together these lemmas yield Theorem 28.

Lemma 25. *The images of maximal pools in SEL_{nk} under products of k-ary Boolean functions are exactly the maximal elements of k-WALKS_n.*

Proof. Let Q be maximal in SEL_{nk} and let $\phi_1, \dots, \phi_n \colon \mathbb{B}^k \to \mathbb{B}$. In Q from one bitstring to the next exactly one position is changed from 0 to 1. Hence, in the image $(\phi_1 \times \dots \times \phi_n)(Q)$ also only one position can, but need not, change. Leaving out consecutive duplicates, the image forms a walk. Note that the walk need not be self-avoiding. Next, consider the transition count of some specific position index i. It is at most k, as a bitflip in the walk at index i can occur only, if a bitflip occurs in the pool Q at one of the positions used by the function ϕ_i – and ϕ_i uses only k positions. Vice versa, it is easily seen that every walk with transition counts at most k is the image of some selective pool. □

Intuitively, if we allow larger transition counts we get longer walks. The following lemma states that this is, indeed, correct.

Lemma 26. *Let $nk + 1 < 2^n$. Then k-$\mathrm{WALKS}_n \subsetneq (k+1)$-$\mathrm{WALKS}_n$.*

Proof. Let $b_1, \dots, b_m \in \mathbb{B}^n$ be a walk with transition counts at most k visiting a maximum number of bitstrings. As $m \leq nk + 1 < 2^n$ there must exist some bitstring $b \in \mathbb{B}^n$ not visited. We extend the walk from b_m as follows: From b_m to b_{m+1} we change the first position where b_m and b differ. Likewise from b_{m+1} to b_{m+2} for the second position and so on. This yields a walk $b_1, \dots, b_m, b_{m+1}, \dots, b_{m+\mathrm{d}(b_m, b)} = b$. This new walk visits *at least one bitstring more than the old walk*, namely b, and has transition counts *at most* $k+1$. Hence $\{b_1, \dots, b_{m+\mathrm{d}(b_m,b)}\} \in (k+1)$-$\mathrm{WALKS}_n \setminus k$-$\mathrm{WALKS}_n$. □

Lemma 27. *Let C be reasonable and $\mathcal{E}' \subsetneq \mathrm{SEL}_{nk}$ be subset closed. Then $C[\mathcal{E}']$ contains only C-cheatable languages.*

Proof. Let $L \in C[\mathcal{E}']$. For input words $w_1, \dots, w_{nk}$ compute pools $P_\sigma \in \mathcal{E}'$ for all *permutations* σ of these words. Each P_σ induces a pool Q_σ for the unpermuted input words. Since $\mathcal{E}'$ misses a maximal selective pool, one of the Q_σ has a size at most nk. Thus, the intersection of the Q_σ has size at most nk. This intersection is a pool for the input words since all Q_σ are. Thus, $L \in C[\text{CHEAT}_{nk}]$. □

Lemma 27′. *Let $\mathcal{E}' \subsetneq \text{SEL}_{nk}$ be subset closed. Then* $\text{RE}[\mathcal{E}'] \subseteq \text{RE}[\text{CHEAT}_{nk}]$.

Theorem 28. *Let C be reasonable. Let N be a C-selective, non-C-cheatable language. Then N is parallel C-supportive. Furthermore,*

$$\mathrm{E}^{\mathrm{L}}_{k\text{-tt}}(N) \not\subseteq \mathrm{R}^{C}_{(k-1)\text{-tt}}\big(C[\text{SEL}]\big) \text{ and}$$
$$\mathrm{E}^{\mathrm{L}}_{k\text{-T}}(N) \not\subseteq \mathrm{R}^{C}_{(2^k-2)\text{-tt}}\big(C[\text{SEL}]\big).$$

Proof. We start with $\mathrm{E}^{\mathrm{L}}_{k\text{-tt}}(N) \not\subseteq \mathrm{R}^{C}_{(k-1)\text{-tt}}\big(C[\text{SEL}]\big)$. Pick some n large enough such that $n(k-1)+1 < 2^n$. Because of Lemma 27, we can apply Theorem 24 to N. It states that the smallest subset closed n-covering $\mathcal{D}$ for which $\mathrm{E}^{\mathrm{L}}_{k\text{-tt}}(N) \subseteq C[\mathcal{D}]$ is k-WALKS$_n$. By Theorem 23 the class $\mathrm{R}^{C}_{(k-1)\text{-tt}}\big(C[\text{SEL}]\big)$ is contained in $C[(k-1)\text{-WALKS}_n]$. If we had $\mathrm{E}^{\mathrm{L}}_{k\text{-tt}}(N) \subseteq \mathrm{R}^{C}_{(k-1)\text{-tt}}\big(C[\text{SEL}]\big)$ then the class $\mathrm{E}^{\mathrm{L}}_{k\text{-tt}}(N)$ would be contained in $C[(k-1)\text{-WALKS}_n]$. As $\mathcal{D}$ is minimal, we would get $(k-1)\text{-WALKS}_n = k\text{-WALKS}_n$ contradicting Lemma 26.

For Turing reductions, repeat the argument for $k' := 2^k - 1$.

Since $\mathrm{E}^{\mathrm{L}}_{k\text{-tt}}(N) \subseteq \mathrm{R}^{C}_{k\text{-tt}}(N)$, the language N is parallel C-supportive. □

Theorem 28′. *Let N be weakly semirecursive and nonrecursive. Then*

$$\mathrm{E}^{\mathrm{L}}_{k\text{-tt}}(N) \not\subseteq \mathrm{R}_{(k-1)\text{-wtt}}\big(\text{RE}[\text{SEL}]\big) \text{ and}$$
$$\mathrm{E}^{\mathrm{L}}_{k\text{-T}}(N) \not\subseteq \mathrm{R}_{(2^k-2)\text{-wtt}}\big(\text{RE}[\text{SEL}]\big).$$

Proof. The proof is the same as the proof of Theorem 28, except for the addition that by the Nonspeedup Theorem we have $\text{RE}[\text{CHEAT}] = \text{REC}$. □

5 Application 1: A Powerful Relativisation

Hemaspaandra et al. [12] proved the relation $\mathrm{E}^{\mathrm{P}}_{(k+1)\text{-tt}}\big(\text{P}[\text{SEL}]\big) \not\subseteq \mathrm{R}^{\mathrm{P}}_{k\text{-tt}}\big(\text{P}[\text{SEL}]\big)$. From this, one easily deduces that P[SEL] is parallel polynomial time supportive. We show that the relation still holds, if we *relativise only the right hand side.*

Theorem 29. *Let A be an oracle. Then* $\mathrm{E}^{\mathrm{P}}_{(k+1)\text{-tt}}\big(\text{P}[\text{SEL}]\big) \not\subseteq \mathrm{R}^{\mathrm{P}^A}_{k\text{-tt}}\big(\text{P}^A[\text{SEL}]\big)$.

Proof. Fix the oracle A. We wish to apply Theorem 28 with $C = \text{P}^A$. In order to do so, we must find a P-selective language N that is not P^A-cheatable. The class P[SEL] is uncountable due to an argument of McLaughlin and Martin, see [13] and [23]. But the class $\text{P}^A[\text{CHEAT}]$ is countable by a relativised version of the Nonspeedup Theorem. Hence, there exists a language $N \in \text{P}[\text{SEL}] \setminus \text{P}^A[\text{CHEAT}]$. For this N, Theorem 28 tells us that $\mathrm{E}^{\mathrm{L}}_{(k+1)\text{-tt}}(N) \not\subseteq \mathrm{R}^{\mathrm{P}^A}_{k\text{-tt}}\big(\text{P}^A[\text{SEL}]\big)$ and hence $\mathrm{E}^{\mathrm{P}}_{(k+1)\text{-tt}}\big(\text{P}[\text{SEL}]\big) \not\subseteq \mathrm{R}^{\mathrm{P}^A}_{k\text{-tt}}\big(\text{P}^A[\text{SEL}]\big)$. □

Corollary 30. *Let C be reasonable and FC countable. Then*

$$C[\text{SEL}] \subsetneq \mathrm{R}^{C}_{1\text{-tt}}\big(C[\text{SEL}]\big) \subsetneq \mathrm{R}^{C}_{2\text{-tt}}\big(C[\text{SEL}]\big) \subsetneq \dots,$$
$$C[\text{SEL}] \subsetneq \mathrm{R}^{C}_{1\text{-parity}}\big(C[\text{SEL}]\big) \subsetneq \mathrm{R}^{C}_{2\text{-parity}}\big(C[\text{SEL}]\big) \subsetneq \dots,$$

and likewise for the equivalence closures.

Proof. Using a dovetail argument, it is easily seen that $FC \subseteq \text{FREC}^A$ for some oracle A. The truth-table hierarchy now follows from the same argument as in the proof of Theorem 29: There are uncountably many C-selective languages, but only countably many REC^A-cheatable languages. The parity hierarchy follows from Lemma 22. $\square$

Note that there also exist hierarchies of the *Turing* reduction closures for all computational models for which Fact 21 holds. Furthermore, a modification of our proofs shows that the Turing hierarchy is proper also in a recursive setting.

6 Application 2: Serial versus Parallel Queries

Fact 21 tells us that if we simulate serial queries to a P-selective language by parallel queries, we cannot avoid an exponential increase in the number of queries. In [12] Hemaspaandra et al. ask whether perhaps we have at least $\text{E}^{\text{P}}_{k\text{-T}}(\text{P}[\text{SEL}]) \subseteq \text{R}^{\text{P}}_{(2^k-2)\text{-tt}}(\text{P}[\text{SEL}])$. This is not the case. Using the same arguments as in the proof of Theorem 29, we even get the following theorem.

Theorem 31. *Let A be an oracle. Then* $\text{E}^{\text{P}}_{k\text{-T}}(\text{P}[\text{SEL}]) \not\subseteq \text{R}^{\text{P}^A}_{(2^k-2)\text{-tt}}(\text{P}^A[\text{SEL}])$.

7 Application 3: Verboseness

The languages in $\text{RE}[m\text{-SIZE}_n]$ are called (m,n)*-verbose* and the languages in $\text{REC}[m\text{-SIZE}_n]$ *strongly* (m,n)*-verbose* [7]. Every language is $(2^n,n)$-verbose. If a language is not $(2^n-1,n)$-verbose, it is called n*-superterse* [7].

Theorem 32. *Let C be reasonable, $FC \subseteq \text{FREC}$ and let m,n,k be positive integers. Then the following statements are equivalent:*

1. *All languages in $\text{E}^C_{k\text{-tt}}(C[\text{SEL}])$ are (m,n)-verbose.*
2. *All languages in $\text{R}^C_{k\text{-tt}}(C[\text{SEL}])$ are (m,n)-verbose.*
3. *Every walk on $\mathbb{B}^n$ with transition counts at most k visits at most m bitstrings.*

Proof. Statement 2 trivially implies statement 1. Statement 3 implies statement 2, since $\text{R}^C_{k\text{-tt}}(C[\text{SEL}]) \subseteq \text{REC}[k\text{-WALKS}_n]$ by Theorem 23. Statement 1 implies statement 3, since by the same argument as in Theorem 28′ the covering $k\text{-WALKS}_n$ is the smallest covering $\mathcal{D}$ such that $\text{R}^C_{k\text{-tt}}(C[\text{SEL}]) \subseteq \text{RE}[\mathcal{D}]$. If we also have $\text{R}^C_{k\text{-tt}}(C[\text{SEL}]) \subseteq \text{RE}[m\text{-SIZE}_n]$, we get $k\text{-WALKS}_n \subseteq m\text{-SIZE}_n$. $\square$

The above theorem reduces the problem of quantifying the amount of verboseness of reduction closures of selectivity classes to the purely combinatorial problem of finding walks visiting a maximum number of bitstrings. The following fact gives an easy upper bound on this number. Beigel et al. [7] conjecture that the bound is tight for all k for which $nk \le 2^n$, and they prove this for $k < 7$. Lemma 34 shows that the bound is also tight for $k = 2^n/n$ and $n = 2^r$. The situation where n is not a power of 2 is discussed, but not fully solved, in [8].

Fact 33 ([7]). *Every walk on $\mathbb{B}^n$ with transition counts at most k visits at most $nk+1$ bitstrings for odd k and nk bitstrings for even k.*

Lemma 34. *For $n = 2^r$ and $r \ge 1$ we have* $(2^n/n)\text{-WALKS}_n \not\subseteq (2^n-1)\text{-SIZE}_n$.

Proof. Wagner and West [25] proved that for $n = 2^r$ there exists a *balanced n-bit Gray code*. But such a code is a self-avoiding n-bit cycle that has transition counts exactly $2^n/n$. □

Corollary 35. *Let $n = 2^r$ for some $r \geq 1$. Then there exists an n-superterse language in $\mathrm{E}^{\mathrm{L}}_{2^n/n\text{-tt}}(\mathrm{L}[\mathrm{SEL}])$, but none in $\mathrm{R}^{\mathrm{REC}}_{(2^n/n-1)\text{-tt}}(\mathrm{REC}[\mathrm{SEL}])$.*

8 Application 4: Frequency Computations

The languages in $\mathrm{REC}_{\mathrm{dist}}[(n-m)\text{-FREQ}_n]$ are called *(m,n)-recursive* [21]. For them, for any n distinct words a bitstring can be computed that agrees with their characteristic string on at least m positions. Arguing as in the previous section for Theorem 32 we get the following theorem.

Theorem 36. *Let C be reasonable and $FC \subseteq \mathrm{FREC}$. Then the following statements are equivalent for $0 < m < n$ and $k \geq 1$:*

1. *All languages in $\mathrm{E}^C_{k\text{-tt}}(C[\mathrm{SEL}])$ are (m,n)-recursive.*
2. *All languages in $\mathrm{R}^C_{k\text{-tt}}(C[\mathrm{SEL}])$ are (m,n)-recursive.*
3. *Every walk on $\mathbb{B}^n$ with transition counts at most k is contained in a closed ball of radius $n-m$.*

The following lemma gives loose upper and lower bounds for k from the above theorem in terms of the *covering number* $\mathrm{k}(n,r)$. It is the smallest number of closed balls of radius r needed to cover $\mathbb{B}^n$. Except for some special cases [10], only upper and lower bounds are known for the covering number.

Lemma 37. *For $0 < m < n$ let $\kappa := \mathrm{k}(n, m-1) - 1$. Then*

$$\kappa\text{-WALKS}_n \not\subseteq (n-m)\text{-FREQ}_n,$$
$$\left\lfloor \frac{\kappa - 1}{n} \right\rfloor\text{-WALKS}_n \subseteq (n-m)\text{-FREQ}_n.$$

Proof. For the first claim, we present a walk on $\mathbb{B}^n$ with transition counts at most κ whose complement does not contain a closed ball of radius $m-1$ and which is hence not contained in a closed ball of radius $n-m$. Let $b_1, \dots, b_{\mathrm{k}(n,m-1)}$ be bitstrings such that $\mathrm{B}_{m-1}(b_1), \dots, \mathrm{B}_{m-1}(b_{\mathrm{k}(n,m-1)})$ cover $\mathbb{B}^n$. Consider a walk starting at b_1. To get from b_1 to b_2 we only need to change every position at most once. Likewise from b_2 to b_3 and so on. We get a walk visiting all b_i with transition counts at most κ. As every ball $\mathrm{B}_{m-1}(b)$ contains at least one b_i, the complement of the constructed walk does not contain any $\mathrm{B}_{m-1}(b)$.
For the second claim let $b_1, \dots, b_\ell$ be an arbitrary walk with transition counts at most $(\kappa-1)/n$. It visits at most $n(\kappa-1)/n + 1 = \kappa$ many bitstrings. We claim that the complement of the walk contains a ball of radius $m-1$. If this were not the case $\mathrm{B}_{m-1}(b_1), \dots, \mathrm{B}_{m-1}(b_\ell)$ would *cover* $\mathbb{B}^n$ which is impossible since $\ell \leq \kappa < \mathrm{k}(n, m-1)$. □

Corollary 38. *There exists a language in $\mathrm{E}^{\mathrm{L}}_{(\mathrm{k}(n,m-1)-1)\text{-tt}}(\mathrm{L}[\mathrm{SEL}])$ which is not (m,n)-recursive, while all languages in $\mathrm{R}^{\mathrm{REC}}_{\lfloor(\mathrm{k}(n,m-1)-2)/n\rfloor\text{-tt}}(\mathrm{REC}[\mathrm{SEL}])$ are.*

As $\mathrm{k}(2r+1, r-1) = 7$ by [10], for all r there exists a language which is 6-tt L-equivalent to an L-selective language, but not $(r, 2r+1)$-recursive. For another example, since $\mathrm{k}(23,3) = 4096$ by [10], every language reducible to a semirecursive language asking 178 truth-table queries is $(4,23)$-recursive.

References

[1] M. Agrawal and V. Arvind. Polynomial time truth-table reductions to P-selective sets. In *Proc. Ninth Annual Structure in Complexity Theory Conference*, pages 24–30. IEEE Computer Society Press, 1994.

[2] R. Beigel. *Query-limited reducibilities.* PhD thesis, Stanford University, Stanford, USA, 1987.

[3] R. Beigel. Bounded queries to SAT and the Boolean hierarchy. *Theoretical Comput. Sci.*, 84(2):199–223, 29 July 1991.

[4] R. Beigel, W. Gasarch, M. Kummer, G. Martin, T. McNicholl, and F. Stephan. The complexity of ODD_n^A. *J. of Symb. Logic*, 65(1):1–18, 2000.

[5] R. Beigel, W. I. Gasarch, and E. Kinber. Frequency computation and bounded queries. *Theoretical Comput. Sci.*, 163(1–2):177–192, 30 Aug. 1996.

[6] R. Beigel, M. Kummer, and F. Stephan. Approximable sets. *Inform. and Computation*, 120(2):304–314, 1995.

[7] R. Beigel, M. Kummer, and F. Stephan. Quantifying the amount of verboseness. *Inform. and Computation*, 118(1):73–90, Apr. 1995.

[8] G. S. Bhat and C. D. Savage. Balanced gray codes. *The Electr. J. of Combinatorics*, 3:#R25, 1996.

[9] H.-J. Burtschick and W. Lindner. On sets Turing reducible to p-selective sets. *ACM Trans. Comput. Syst.*, 30(2):135–143, 1997.

[10] G. Cohen, A. Lobstein, and N. Sloane. Further results on the covering radius of codes. *IEEE Trans. Inform. Theory*, 32:680–694, 1986.

[11] L. A. Hemachandra. The strong exponential hierarchy collapses. *J. Comput. Syst. Sci.*, 39(3):299–322, Dec. 1989.

[12] L. A. Hemaspaandra, A. Hoene, and M. Ogihara. Reducibility classes of P-selective sets. *Theoretical Comput. Sci.*, 155(2):447–457, 1996.

[13] C. G. Jockusch, Jr. Semirecursive sets and positive reducibility. *Transactions of the American Mathematical Society*, 131:420–436, 1968.

[14] C. G. Jockusch, Jr. and J. C. Owings, Jr. Weakly semirecursive sets. *J. of Symb. Logic*, 55(2):637–644, June 1990.

[15] K.-I. Ko. On the definitions of some complexity classes of real numbers. *Math. Syst. Theory*, 16(2):95–109, May 1983.

[16] R. E. Ladner, N. A. Lynch, and A. L. Selman. A comparison of polynomial time reducibilities. *Theoretical Comput. Sci.*, 1(2):103–123, Dec. 1975.

[17] A. Nickelsen. On $\mathcal{D}$-verbose languages. In *Proc. STACS 97*, volume 1200 of *LNCS*, pages 307–318. Springer-Verlag, 1997.

[18] A. Nickelsen. *Polynomial Time Partial Information Classes.* PhD thesis, Technische Universität Berlin, Germany, 1999.

[19] P. Odifreddi. *Classical Recursion Theory*, volume 125 of *Studies in Logic and the Foundations of Mathematics.* North-Holland, 1989.

[20] M. Ogihara. Polynomial-time membership comparable sets. *SIAM J. Comput.*, 24(5):1068–1081, Oct. 1995.

[21] G. F. Rose. An extended notion of computability. In *Proc. 1960 Int. Congress for Logic, Methodology, and Philosophy of Science*, 1960.

[22] A. L. Selman. P-selective sets, tally languages, and the behavior of polynomial time reducibilities on NP. *Math. Syst. Theory*, 13:55–65, 1979.

[23] A. L. Selman. Some observations on NP real numbers and P-selective sets. *J. Comput. Syst. Sci.*, 23(3):326–332, Dec. 1981.

[24] T. Tantau. Combinatorial representations of partial information classes and their truth-table closures. Master's thesis, Technische Universität Berlin, available at `http://www.eccc.uni-trier.de/eccc`, Germany, Aug. 1999.

[25] D. G. Wagner and J. West. Construction of uniform gray codes. *Congressus Numerantium*, 80:217–223, 1991.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'