Symmetric Alternation Captures **BPP**

Alexander Russell[*]
Department of Mathematics

Ravi Sundaram[†]
Laboratory for Computer Science

Massachusetts Institute of Technology
Cambridge, MA 02139

November 14, 1995

Abstract

We introduce the natural class $\mathbf{S}_2^P$ containing those languages which may be expressed in terms of two *symmetric quantifiers*. This class lies between Δ_2^P and $\Sigma_2^P \cap \Pi_2^P$ and naturally generates a "symmetric" hierarchy corresponding to the polynomial-time hierarchy. We demonstrate, using the probabilistic method, new containment theorems for **BPP**. We show that **MA** (and hence **BPP**) lies within $\mathbf{S}_2^P$, improving the constructions of [10, 8] (which show that $\mathbf{BPP} \subset \Sigma_2^P \cap \Pi_2^P$). Symmetric alternation is shown to enjoy two strong structural properties which are used to prove the desired containment results. We offer some evidence that $\mathbf{S}_2^P \neq \Sigma_2^P \cap \Pi_2^P$ by demonstrating an oracle so that $\mathbf{S}_2^{P,O} \neq \Sigma_2^{P,O} \cap \Pi_2^{P,O}$ assuming that the machines make only "positive" oracle queries.

1 Introduction

Since the inclusion of randomness among those resources for which we have agreeable computational models, determination of the exact relationship between randomness and other such computational resources has become a major project. The relationship between *space* and randomness, elucidated by startling pseudorandom constructions ([1, 9]), is remarkably well understood. These constructions demonstrate that space-bounded computation benefits little by the use of randomness. The analogous relationship with *time* has proved

[*]E-mail address: `acr@theory.lcs.mit.edu`. Supported by an NSF Graduate Fellowship and grants NSF 92-12184, AFOSR F49620-92-J-0125, and DARPA N00014-92-1799

[†]E-mail address: `koods@theory.lcs.mit.edu`. Supported by grants NSF 92-12184, AFOSR F49620-92-J-0125, and DARPA N00014-92-J-1799

less tractable: the only (non-trivial) relationships depend on unproven complexity-theoretic assumptions ([7, 12]). We continue the study initiated by Sipser [10] of the relationship between randomness and *quantification*, that is, the relationship between **BPP** and classes arising by appropriate quantification of polynomial-time predicates (e.g. **NP**, **coNP** and other classes in the polynomial-time hierarchy [11]).

BPP was first shown to lie in the polynomial-time hierarchy in [10] which demonstrates that $\mathbf{BPP} \subseteq \Sigma_2^P$ (and hence that $\mathbf{BPP} \subset \Sigma_2^P \cap \Pi_2^P$). We introduce a natural quantified class, $\mathbf{S}_2^P$, and demonstrate that

$$\mathbf{BPP} \subset \mathbf{S}_2^P \subset \Sigma_2^P \cap \Pi_2^P.$$

Given the unnaturality of $\Sigma_2^P \cap \Pi_2^P$, the naturality of $\mathbf{S}_2^P$ suggests that $\mathbf{S}_2^P \subsetneq \Sigma_2^P \cap \Pi_2^P$ and hence that this containment is more than a solely conceptual improvement.

The class $\mathbf{S}_2^P$ consists of those languages $\mathcal{L}$ which may be decided by a polynomial-time machine that receives counsel from two provers in such a manner that when the input $x \in \mathcal{L}$ there is a "witness" w_1 which the first prover may provide so that regardless of the information provided by the second prover, the machine accepts and, similarly, when $x \notin \mathcal{L}$, there is a "witness" w_2 which the second prover may supply so that, regardless of the information supplied by the first prover, the machine rejects. The special sort of alternation involved here we refer to as *symmetric alternation.* The $\mathbf{S}_2$ operator enjoys some remarkable structural properties:

- $\mathbf{S}_2 \cdot \mathbf{BP} \cdot \mathbf{P} \subset \mathbf{S}_2 \cdot \mathbf{P}$,
- $\mathbf{P}^{(\mathbf{S}_2 \cdot P)} = \mathbf{S}_2 \cdot P$.

We use these structural properties to conclude that

- $\mathbf{BPP} \subset \mathbf{MA} \subset \mathbf{S}_2^P$, and
- $\Delta_2^P = \mathbf{P}^{\mathbf{NP}} \subset \mathbf{S}_2^P$.

Considering the strong structural properties which $\mathbf{S}_2^P$ enjoys, it seems unlikely that $\mathbf{S}_2^P = \Sigma_2^P \cap \Pi_2^P$. In light of the above containment theorems, however, we would like to present as much evidence as possible in this direction. One standard method of offering evidence that two classes are different is to demonstrate an oracle which separates them. In §3 we construct an oracle which separates S_2^P from $\Sigma_2^P \cap \Pi_2^P$ under the assumption that the machines involved are monotone. The framework we develop to build this oracle can be used to simplify the construction given by [3] of an oracle separating Σ_2^P and Π_2^P.

2 Definitions and Containment Results

Σ is used to denote the alphabet and may be assumed to be $\{0, 1\}$ without loss of generality. Throughout, the variable n denotes $|w|$, the length of the input in question. For $m \in \mathbb{N}$, we use $\exists^m x$ as shorthand for $\exists x(|x| = m)$, $\exists^m!x$ if such x is unique. $\forall^m x$ and $\forall^m!x$ is similarly used.

Definition 2.1 **($\mathbf{S}_2 \cdot \mathfrak{C}$)** *For a complexity class $\mathfrak{C}$ we define $\mathbf{S}_2 \cdot \mathfrak{C}$ to be the complexity class consisting of those languages $\mathcal{L}$ for which there exists $\mathcal{C} \in \mathfrak{C}$ and a polynomial q so that*

- $w \in \mathcal{L} \Rightarrow \exists^{q(n)}x, \forall^{q(n)}y, \langle w, x, y\rangle \in \mathcal{C}$, *and*
- $w \notin \mathcal{L} \Rightarrow \exists^{q(n)}y, \forall^{q(n)}x, \langle w, x, y\rangle \notin \mathcal{C}$.

Notice that the acceptance criteria for the $\mathbf{S}_2$ operator has the form of the acceptance criteria for Σ_2^P. The rejection criteria is similarly related to that of Π_2^P. $\mathbf{S}_2 \cdot \mathbf{P}$, then, is clearly inside $\Sigma_2^P \cap \Pi_2^P$. Notice that $\mathbf{S}_2^P$, like **BPP** and **IP**, is a *promise* class – the criteria for acceptance and rejection are not complements of each other.

Definition 2.2 **($\mathbf{S}_2^P$)** $\mathbf{S}_2^P \stackrel{\text{def}}{=} \mathbf{S}_2 \cdot \mathbf{P}$.

Definition 2.3 **($\mathbf{S}_{2k}^P$)** $\mathbf{S}_{2k}^P \stackrel{\text{def}}{=} \overbrace{\mathbf{S}_2 \cdot \mathbf{S}_2 \cdots \mathbf{S}_2}^{k} \cdot \mathbf{P}$.

Theorem 2.4 $\Sigma_1^P \bigcup \Pi_1^P \subset \mathbf{S}_2^P \subset \Sigma_2^P \bigcap \Pi_2^P$.

Proof: $\Sigma_1^P \subset \mathbf{S}_2^P \subset \Sigma_2^P$ and $\mathbf{S}_2^P$ is closed under complement. $\Box$

Corollary 2.5 $\Sigma_k^P \bigcup \Pi_k^P \subset \mathbf{S}_{2k}^P \subset \Sigma_{2k}^P \bigcap \Pi_{k2}^P$.

This allows us to conclude that $\mathbf{PH} \stackrel{\text{def}}{=} \bigcup_k \Sigma_2^P = \bigcup_k \mathbf{S}_{2k}^P$, so that $\mathbf{S}_{2k}^P$ for $k = 1, 2 \ldots$ form a hierarchy which collapses if and only if the polynomial hierarchy collapses.

Theorem 2.6 $\mathbf{P}^{\mathbf{S}_2^P} \subset \mathbf{S}_2^P$

Proof: Let $\mathfrak{S} \in \mathbf{S}_2^P$ and $S(\cdot, \cdot, \cdot)$ be a polynomial time machine accepting $\mathfrak{S}$ according to the definition of $\mathbf{S}_2^P$. Let $\mathcal{L} \in \mathbf{P}^{\mathfrak{S}}$ and let $D^{\mathfrak{S}}$ be a deterministic polynomial-time machine deciding $\mathcal{L}$. A *computation suggestion* of $D^{\mathfrak{S}}(w)$ consists of a sequence of pairs $(\mathfrak{d}_i, \mathfrak{a}_i)$ for $i = 1, \ldots, t$ so that

- each $\mathfrak{d}_i$ is an instantaneous description (see [6]) of $D^{\mathfrak{S}}$,
- $\mathfrak{d}_1$ is the initial instantaneous description of $D^{\mathfrak{S}}(w)$,
- $\mathfrak{d}_t$ is a final instantaneous description of $D^{\mathfrak{S}}$,
- if $\mathfrak{d}_i$ does not find $D^{\mathfrak{S}}$ in its query state, then $\mathfrak{d}_i \vdash_D \mathfrak{d}_{i+1}$,
- if $\mathfrak{d}_i$ finds $D^{\mathfrak{S}}$ querying q_i, then there is a response r_i so that $\mathfrak{d}_i \vdash_D \mathfrak{d}_{i+1}$ with response r_i and $\mathfrak{a}_i$ is a string of length appropriate for the quantified inputs of S on input w.

We construct a machine $T(\cdot,\cdot,\cdot)$ accepting $\mathcal{L}$ according to the definition of $\mathbf{S}_2^P$. $T(w,x,y)$ first examines x, rejecting unless x is a computation suggestion for $D^{\mathfrak{S}}$ so that $x = (\mathfrak{d}_i^x, \mathfrak{a}_i^x)_{i \le t_x}$. T then examines y, accepting unless y is a computation suggestion for $D^{\mathfrak{S}}$ (so that $y = (\mathfrak{d}_i^y, \mathfrak{a}_i^y)_{i \le t_y}$). If $t_x = t_y$ and $\forall i \in \{1, \ldots t_x\}$ we have that $\mathfrak{d}_i^x = \mathfrak{d}_i^y$ then T accepts exactly when $\mathfrak{d}_{t_x}^x = \mathfrak{d}_{t_y}^y$ is an accepting description. Otherwise there is i_0 so that $\mathfrak{d}_{i_0}^x = \mathfrak{d}_{i_0}^y$ but $\mathfrak{d}_{i_0+1}^x \neq \mathfrak{d}_{i_0+1}^y$. Evidently, $\mathfrak{d}_{i_0}^x$ is a query state of $D^{\mathfrak{S}}$ (otherwise there is a unique next state, upon which both x and y must agree if they are computation suggestions). Let q_{i_0} be the query appearing in this description. Assume without loss of generality that the computation suggestion of x claims that $q_{i_0} \in \mathfrak{S}$. T then simulates $S(q_{i_0}, \mathfrak{a}_{i_0}^x, \mathfrak{a}_{i_0}^y)$ accepting exactly when S accepts. Notice that for input w, there is a *correct* computation suggestion $\Delta_w = (\mathfrak{d}_i, \mathfrak{a}_i)_{i \le t}$ (that is, one in which every oracle query is answered correctly) so that for each query q_i,

- if $q_i \in \mathfrak{S}$ then $\forall z, S(q_i, \mathfrak{a}_i, z)$ accepts, and
- if $q_i \notin \mathfrak{S}$ then $\forall z, S(q_i, z, \mathfrak{a}_i)$ rejects.

Then

- if $w \in \mathcal{L}$, then $\forall^{q(n)} y, T(w, \Delta_w, y)$ accepts, and
- if $w \notin \mathcal{L}$, then $\forall^{q(n)} x, T(w, x, \Delta_w)$ rejects,

as desired. $\square$

Corollary 2.7 $\Delta_2^P = \mathbf{P}^{\mathbf{NP}} \subset \mathbf{P}^{\mathbf{S}_2^P} \subset \mathbf{S}_2^P$.

Corollary 2.8 $\Delta_{k+1}^P \subset \mathbf{S}_{2k}^P$.

The proof of Theorem 2.9 below is a generalization of the argument of Lautemann [8].

Theorem 2.9 $\mathbf{S}_2 \cdot \mathbf{BP} \cdot \mathbf{P} \subset \mathbf{S}_2 \cdot \mathbf{P}$.

Proof: Let $\mathcal{L} \in \mathbf{S}_2 \cdot \mathbf{BP} \cdot \mathbf{P}$. Let $\mathcal{D} \in \mathbf{P}$ and q, r be polynomials such that

- $w \in \mathcal{L} \implies \exists^{q(n)} x, \forall^{q(n)} y, \Pr_{r \in_R \Sigma^{r(n)}}[\langle w, x, y, r\rangle \in \mathcal{D}] \geq 1 - 2^{-q(n)-n}$
- $w \notin \mathcal{L} \implies \exists^{q(n)} y, \forall^{q(n)} x, \Pr_{r \in_R \Sigma^{r(n)}}[\langle w, x, y, r\rangle \in \mathcal{D}] \leq 2^{-q(n)-n}$

Fix $w \in \Sigma^*$ and let $\hat{x} \in \Sigma^{q(n)}$ be so that for all $y \in \Sigma^{q(n)}$, $\Pr_{r \in_R \Sigma^{r(n)}}[\langle w, \hat{x}, y, r\rangle \in \mathcal{D}] \geq 1 - 2^{-q(n)-n}$. Let $\mathcal{W}_y \subset \Sigma^{r(n)}$ be the collection of random strings r for which $\langle w, \hat{x}, y, r\rangle \in \mathcal{D}$ and let $\mathcal{W} \stackrel{\text{def}}{=} \bigcap_y \mathcal{W}_y$ (these are the random strings r such that $\langle w, \hat{x}, y, r\rangle \in \mathcal{D}$ for all y). For a set B let $\mu(B)$ denote the measure of the set. Then $\forall y, \mu(\mathcal{W}_y) \geq 1 - 2^{-q(n)-n}$ so that $\mu(\mathcal{W}) \geq 1 - 2^{-n}$. As in [8], we demonstrate that there exists a set $\{\sigma \stackrel{\text{def}}{=} \sigma_1, \ldots, \sigma_{r(n)}\}$ of elements of $\Sigma^{r(n)}$ so that for every $\tau \in \Sigma^{r(n)}$, there is some i so that $\sigma_i \oplus \tau \in \mathcal{W}$ ($\oplus$ stands for the binary operator that returns the bitwise XOR of the operands). Selecting $\sigma_1, \ldots, \sigma_{r(n)}$

uniformly and independently at random from $\Sigma^{r(n)}$, let $\mathcal{B}_\tau$ be the event that for each i, $\sigma_i \oplus \tau \notin \mathcal{W}$. Then $\Pr[\mathcal{B}_\tau] \leq 2^{-nr(n)}$ so that

$$\Pr[\bigvee_{\tau \in \Sigma^{r(n)}} \mathcal{B}_\tau] \leq \sum_\tau \Pr[\mathcal{B}_\tau] = 2^{r(n)} 2^{-nr(n)} = 2^{r(n)(1-n)} < 1.$$

Hence there is a set $\{\sigma = \sigma_1, \ldots, \sigma_{r(n)}\}$ so that for all τ, there is i so that $\sigma_i \oplus \tau \in \mathcal{W}$.

Suppose now that $w \notin \mathcal{L}$. There is then $\hat{y} \in \Sigma^{q(n)}$ so that for all $x \in \Sigma^{q(n)}$, $\Pr_r[\langle w, x, \hat{y}, r\rangle \in \mathcal{D}] \leq 2^{-q(n)-n}$. As before, let $\mathcal{W}_x \stackrel{\text{def}}{=} \{r \in \Sigma^{r(n)} \mid \langle w, x, \hat{y}, r\rangle \in \mathcal{D}\}$. Define $\mathcal{W} \stackrel{\text{def}}{=} \bigcup_x \mathcal{W}_x$ so that $\mu(\mathcal{W}) \leq 2^{-n}$. Then we show that there is a set $\tau \stackrel{\text{def}}{=} \{\tau_1, \ldots, \tau_{r(n)^2}\}$ of elements of $\Sigma^{q(n)}$ so that for all $\sigma = \{\sigma_1, \ldots, \sigma_{r(n)}\}$, $\exists j, \forall i, \sigma_i \oplus \tau_j \notin \mathcal{W}$. Selecting $\tau_1, \ldots, \tau_{r(n)^2}$ independently and uniformly at random, let $\mathcal{B}_\sigma$ be the event that $\forall j, \exists i, \sigma_i \oplus \tau_j \in \mathcal{W}$. Then

$$\Pr[\mathcal{B}_\sigma] \leq \prod_j \sum_{i=1}^{r(n)} \Pr[\sigma_i \oplus \tau_j \in \mathcal{W}] = \prod_j \sum_{i=1}^{r(n)} 2^{-n} = (\frac{r(n)}{2^n})^{r(n)^2}.$$

Hence,

$$\Pr[\bigvee_\sigma \mathcal{B}_\sigma] \leq \sum_\sigma \Pr[\mathcal{B}_\sigma] = 2^{r(n)^2} \left(\frac{r(n)}{2^n}\right)^{r(n)^2} < 1.$$

Therefore, there is a set $\{\tau_1, \ldots, \tau_{r(n)^2}\}$ with the property that for any set $\{\sigma_1, \ldots, \sigma_{r(n)}\}$, there is some j so that $\sigma_i \oplus \tau_j \notin \mathcal{W}$ for all i. In light of this, consider the deterministic polynomial time machine $M(\cdot, \cdot, \cdot)$ which, on input (w, α, β),

- checks the format of α, rejecting unless $\alpha = \langle x; \sigma_1, \ldots, \sigma_{r(n)}\rangle$,
- checks the format of β, accepting unless $\beta = \langle y; \tau_1, \ldots, \tau_{r(n)^2}\rangle$, and
- accepts iff for each τ_j, there is some σ_i so that $\langle w, x, y, \sigma_i \oplus \tau_j\rangle \in \mathcal{D}$.

From above, if $x \in \mathcal{L}$ then setting α to be the $\langle \hat{x}; \sigma_1, \ldots, \sigma_{r(n)}\rangle$ promised in the first part of the above discussion we have that D accepts regardless of β. Similarly, if $x \notin \mathcal{L}$ then setting β to be the $\langle \hat{y}; \tau_1, \ldots, \tau_{r(n)^2}\rangle$ promised in the second part of the above discussion, we have that D rejects regardless of α. □

Corollary 2.10 **MA** $\subset \mathbf{S}_2^P$.

Corollary 2.11 **BPP** $\subset \mathbf{S}_2^P$

3 An Oracle Separating monotone $\mathbf{S}_2^P$ and monotone $\Sigma_2^P \cap \Pi_2^P$

Definition 3.1 *We shall call an oracle Turing machine $M^O(\cdot, \ldots, \cdot)$* monotone *if for any inputs of the machine $x_1, \ldots, x_n$,*

$$O_1 \subseteq O_2 \wedge M^{O_1}(x_1, \ldots, x_n) \textit{ accepts} \implies M^{O_2}(x_1, \ldots, x_n) \textit{ accepts}.$$

We then define $\mathbf{mS}_2^{P,O}$ to be the class of languages accepted by some monotone machine according to the $\mathbf{S}_2^P$ acceptance rules with oracle O. $\mathbf{m}\Sigma_2^{P,O}$ and $\mathbf{m}\Pi_2^{P,O}$ are defined similarly.

The above conclusion that $\mathbf{BPP} \subset \mathbf{S}_2^P \subset \Sigma_2^P \cap \Pi_2^P$ is interesting commensurate with the extent to which we believe that the inclusion $\mathbf{S}_2^P \subset \Sigma_2^P \cap \Pi_2^P$ is strict. We offer evidence for the strictness of this inclusion by demonstration of an oracle O so that $\mathbf{mS}_2^{P,O} \subsetneq \mathbf{m}\Sigma_2^{P,O} \cap \mathbf{m}\Pi_2^{P,O}$.

We begin with some definitions relevant to our construction. For $n \geq 0$, consider subsets T of Σ^{2n} satisfying the Π_2^P predicate $\forall^n x \exists^n y, xy \in T$. This predicate is monotone. Collect together the minterms to form

$$\mathfrak{T} = \{T \subset \Sigma^{2n} \mid \forall^n x \exists^n ! y, xy \in T\}.$$

This set has size 2^{n2^n}. Given a family of minterms $\mathcal{T} \subset \mathfrak{T}$ and a set $W \subset \Sigma^{2n}$ we define

$$\mathcal{T}_W \stackrel{\text{def}}{=} \{T \in \mathcal{T} \mid W \subset T\}$$

which we shall call $\mathcal{T}$ *pinched at* W. A family $\mathcal{T} \subset \mathfrak{T}$ is said to be ϵ*-concentrated at* w if $\Pr_{T \in \mathfrak{T}}[w \in T] \geq \epsilon$. We shall say that $\mathcal{T} \neq \emptyset$ is ϵ*-diffuse on* S if for all $w \in S$, $\mathcal{T}$ is not ϵ-concentrated at w. A family $\mathcal{T}$ which is ϵ-concentrated at τ may be *pinched* at $\{\tau\}$, resulting in $\mathcal{T}_{\{\tau\}}$, with $|\mathcal{T}_{\{\tau\}}| \geq \epsilon |\mathcal{T}|$. A ϵ*-concentration sequence* for a family $\mathcal{T}$ is a sequence $\tau_1, \ldots, \tau_r$ such that

- $\mathcal{T}$ is ϵ-concentrated at τ_1,
- for $i \in \{1, \ldots, r-1\}$, $\mathcal{T}_{\{\tau_1, \ldots, \tau_i\}}$ is ϵ-concentrated at τ_{i+1},
- and $\mathcal{T}_{\{\tau_1, \ldots, \tau_r\}}$ is ϵ-diffuse on $\Sigma^{2n} - \{\tau_1, \ldots, \tau_r\}$.

Lemma 3.2 *Let $\tau_1, \ldots, \tau_r$ be an ϵ-concentration sequence for $\mathcal{T} \subset \mathfrak{T}$. Then*

$$r \leq \frac{-\log \mu(\mathcal{T})}{\log \epsilon + n}$$

where $\mu(\mathcal{T})$ is the density of $\mathcal{T}$ in $\mathfrak{T}$.

Proof: Let $\tau = \{\tau_1, \ldots, \tau_r\}$. Since $\mathcal{T} \subset \mathfrak{T}$, $\mathcal{T}_\tau \subset \mathfrak{T}_\tau$ so that

$$\epsilon^r \mu(\mathcal{T}) \leq \mu(\mathcal{T}_\tau) \leq \mu(\mathfrak{T}_\tau) = 2^{-nr}$$

and hence

$$r \leq \frac{-\log \mu(\mathcal{T})}{\log \epsilon + n}$$

as desired. $\Box$

Theorem 3.3 *There exists an oracle O so that $\mathbf{mS}_2^{P,O} \subsetneq \mathbf{m}\Sigma_2^{P,O} \cap \mathbf{m}\Pi_2^{P,O}$.*

Proof: For a pair of oracles $O_1, O_2 \subset \Sigma^*$, define $O_1 \oplus O_2 \stackrel{\text{def}}{=} \{0x \mid x \in O_1\} \cup \{1y \mid y \in O_2\}$. Let $\mathcal{C}$ be the set of all oracles $O = O_1 \oplus O_2$ so that $\forall n \geq 0$,

$$\exists^n x \forall^n y, xy \in O_1 \iff \forall^n x \exists^n y, xy \in O_2.$$

Let $\mathcal{A}_n = \{C_0 \oplus C_1 \in \mathcal{C} \mid \exists^n x, \forall^n y, xy \in C_0\}$. Define $\mathcal{L}(O_1 \oplus O_2) = \{1^n \mid \exists^n x \forall^n y, xy \in O_1\}$ and notice that for $O \in \mathcal{C}$, we have that $\mathcal{L}(O) \in \mathbf{m}\Sigma_2^{P,O} \cap \mathbf{m}\Pi_2^{P,O}$. We shall construct an oracle $C = C_1 \oplus C_2 \in \mathcal{C}$ so that $\mathcal{L}(C) \notin \mathbf{m}S_2^{P,C}$. Let $\{D_i(\cdot,\cdot,\cdot)\}$ be an enumeration of monotone oracle-S_2^P machines so that for every O, $\mathbf{m}S_2^{P,O} = \{L(D_i^O)\}$. Define $Q_i(n)$ to be the maximum size, over all oracles, x values, and y values, of any query made by $D_i(1^n, x, y)$. C shall be constructed in stages $C_1 \subset C_2 \subset \cdots$ so that $C = \bigcup_i C_i$, stage i constructed to foil a specific monotone S_2^P machine. We shall have that

- for $i < j$, $C_i \subset C_j$ and $C_j - C_i \subset \Sigma^{k_j}$ for some k_j,
- $i < j \implies k_i < k_j$,
- for any oracle O with $O \cap \Sigma^{\leq k_s} = C_s$, $\mathcal{L}(O) \neq L(D_i^O)$ for any $i \leq s$.

Assume we have constructed the first $t-1$ stages, thus defining the oracle to length k_{t-1} and foiling the first $t-1$ machines. We shall construct C_t, foiling D_t. Let $p(n)$ be the running time of D_t. Select n so that $2n > k_{t-1}$, $2n > Q_{t-1}(k_{t-1})$, and $2^n > 2p(n)$. Set $k_t = 2n$. Assume, for contradiction, that regardless of our choice of $C_t \in \mathcal{C}$ (with $C_t \cap \Sigma^{\leq k_{t-1}} = C_{t-1}$), $D_t^{C_t}(1^n, \cdot, \cdot)$ accepts exactly when $C_t \in \mathcal{A}_n$.

For each $u \in \Sigma^n$, let $S_u \stackrel{\text{def}}{=} \{uv \mid |v| = n\}$ and consider the family of oracles

$$\{C_{t-1} \cup S_u \oplus T \mid T \in \mathfrak{T}\}.$$

Associate with each oracle O in this family an appropriate x so that $\forall y, D_t^O(1^n, x, y)$ accepts. There are at most $2^{p(n)}$ various values for x, so some x_u is associated with a fraction of this family of density at least $2^{-p(n)}$. Let $\mathcal{F}(u)$ be the (sub) family so associated with this x_u. Then define $\mathcal{T}(u) = \{T \mid C_{t-1} \cup S_u \oplus T \in \mathcal{F}_u\}$. Let $\tau(u)_1, \tau(u)_2, \ldots, \tau(u)_{t(u)}$ be a $p(n)^{-1}$-concentration sequence for $\mathcal{T}(u)$ and $\tau(u) = \{\tau(u)_1, \tau(u)_2, \ldots, \tau(u)_{t(u)}\}$. By lemma 3.2 the length of this sequence, $t(u)$, is at most

$$\frac{-\log \mu(\mathcal{T}(u))}{\log p(n)^{-1} + n} \leq \frac{p(n)}{n - \log p(n)} \stackrel{(*)}{\leq} p(n)$$

where the inequality $\stackrel{(*)}{\leq}$ follows because $2^n > 2p(n)$. Reiterating, for each $u \in \Sigma^n$, we have selected a family of minterms $\mathcal{F}(u)$ all associated with a certain x_u and a $p(n)^{-1}$-concentration sequence $\tau(u)$ for $\mathcal{T}(u)$.

We similarly construct such sets of maxterms. For a subset $X \subset \Sigma^{2n}$, let $\hat{X} \stackrel{\text{def}}{=} \Sigma^{2n} - X$ be the relative complement of X. For each $v \in \Sigma^n$, consider

$$\{C_{t-1} \cup \hat{T} \oplus \hat{S}_v \mid T \in \mathfrak{T}\}.$$

As above, let y_v be associated with a family of these maxterms $\mathcal{G}(v) \stackrel{\text{def}}{=} \{C_{t-1} \cup \hat{T} \oplus \hat{S}_v \mid T \in \mathcal{T}(v)\}$ so that $\mu(\mathcal{T}(v)) \geq 2^{-p(n)}$ (so that for any oracle O in this set and any x, $D_t^O(1^n, x, y_v)$ rejects). Let $\tau(v)_1, \ldots, \tau(v)_{t(v)}$ be a $p(n)^{-1}$-concentration sequence for $\mathcal{T}(v)$ and $\tau(v) = \{\tau(v)_1, \ldots, \tau(v)_{t(v)}\}$. Again $t(v) \leq p(n)$.

Selecting u and v uniformly and independently at random from Σ^n, we have that

$$\Pr_{u,v}[\exists i, \tau(u)_i \in S_v \text{ or } \exists j, \tau(v)_j \in S_u] \leq \frac{t(u)+t(v)}{2^n} \leq \frac{2p(n)}{2^n} < 1$$

so that there exists a pair $(\tilde{u}, \tilde{v})$ with the property that

- $\forall i, \tau(\tilde{u})_i \notin S_v$ and
- $\forall j, \tau(\tilde{v})_j \notin S_u$.

Notice that an oracle O selected from $\mathcal{F}(\tilde{u}) \cup \mathcal{G}(\tilde{v})$ is accepted by $D_t^O(1^n, x_{\tilde{u}}, y_{\tilde{v}})$ exactly when $O \in \mathcal{F}(\tilde{u})$. Let $M = \Sigma^{2n} - \tau(\tilde{v}) \oplus \tau(\tilde{u})$ and consider the behavior of $D_t^M(1^n, x_{\tilde{u}}, y_{\tilde{v}})$. Suppose that $D_t^M(1^n, x_{\tilde{u}}, y_{\tilde{v}})$ accepts. Since D is monotone, every oracle $O \in \{C_{t-1} \cup \hat{T} \oplus \hat{S}_{\tilde{v}} \mid T \in \mathcal{T}(\tilde{v})_{\tau(\tilde{v})}\}$ must disagree with M in one of the at most $p(n)$ places which $D_t^M(1^n, x_{\tilde{u}}, y_{\tilde{v}})$ queries, which contradicts that $\mathcal{T}(\tilde{v})_{\tau(\tilde{v})}$ is $p(n)^{-1}$-diffuse on $\Sigma^{2n} - \tau(\tilde{v})$. The case when $D_t^M(1^n, x_{\tilde{u}}, y_{\tilde{v}})$ rejects is handled dually. □

4 Conclusions and Open Problems

In this note we studied the notion of *symmetric alternation* by defining the complexity class $\mathbf{S}_2^P$. We observed certain structural properties of the $\mathbf{S}_2$ operator which allowed us to prove some interesting containment results. We were able to show that $\mathbf{BPP} \subset \mathbf{S}_2^P$ using a clever twist of Lautemann's proof [8].

The original motivation for defining and studying the notion of symmetric alternation was a question posed by Uriel Feige. [4] and [5] study situations, in an interactive setting, where the provers do not have complete access to each other's strategies. As a step towards characterizing the class of languages accepted by such interactive proof systems we, in this paper, decided to formalize and study the associated non-interactive version.

It is an interesting open problem to construct an oracle separating $\mathbf{S}_2^P$ and $\Sigma_2^P \cap \Pi_2^P$. This would provide more evidence of the fact that the two classes are indeed different. The interactive version of symmetric alternation, the original motivation for this study, has not been characterized exactly and continues to remain an area of active study.

References

[1] M. Ajtai, J. Komlos, and E. Szemeredi. Deterministic simulation in logspace. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, pages 132–140, 1987.

[2] N. Alon and J. H. Spencer. *The Probabilistic Method.* John Wiley & Sons, Inc., 1992.

[3] T. P. Baker and A. L. Selman. A second step toward the polynomial hierarchy. *Theoretical Computer Science*, 8:177–187, 1979.

[4] U. Feige, A. Shamir, and M. Tennenholtz. The noisy oracle problem. In *Proceedings of CRYPTO 1988*, pages 284–296, 1988.

[5] J. Feigenbaum, D. Koller, and P. Shor, November 1994. Unpublished Manuscript.

[6] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley Series in Computer Science. Addison-Wesley, Reading, Massachusetts, 1979.

[7] R. Impagliazzo, L. A. Levin, and M. Luby. Pseudo-random generation from one-way functions. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, pages 12–24, Seattle, Washington, May 1989.

[8] C. Lautemann. BPP and the polynomial hierarchy. *Information Processing Letters*, 17:215–217, 1983.

[9] N. Nisan and D. Zuckerman. More deterministic simulation in logspace. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, pages 235–244, 1993.

[10] M. Sipser. A complexity theoretic approach to randomness. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, pages 330–335, 1983.

[11] L. J. Stockmeyer. The polynomial time hierarchy. *Theoretical Computer Science*, 3:1–22, 1976.

[12] A. C. Yao. Theory and applications of trapdoor functions. In *23rd Annual Symposium on Foundations of Computer Science*, pages 80–91. IEEE, 1982.