Sharply Bounded Alternation within P

Judy Goldsmith[1] and Stephen A. Bloch[2]
Technical Report # 92-04

October 11, 1995

[1]Computer Science Department, The University of Manitoba, Winnipeg, Manitoba, Canada, R3T 2N2. Project sponsored in part by NSERC operating grant OGP0121527.

[2]Math Department, University of California-San Diego. Project sponsored in part by grants from the University of Manitoba.

1 Introduction

Since the class, P, of polynomial-time recognizable sets was formally introduced in 1965 [6, 9, 20] many theorists have argued that it is a natural formulation of the notion of feasible computation [8]. This argument is supported by the robustness of P (it is essentially machine-model invariant; it is closed under composition and Boolean combinations), and by its plethora of logical characterizations (see [3, 4, 13, 14, 16, 10, 17, 18, 23, 23, 25, 30], for example). Unfortunately, it is too big a class to practically be considered feasible. Programmers are not interested in programs that run in time $\mathcal{O}(n^{17})$, for instance. We propose a hierarchy of complexity classes, the Sharply Bounded Hierarchy (SBH), within P, which we claim represent a better approximation of a programmer's notion of feasibly computable. We discuss the structure of this hierarchy, give examples of problems in the hierarchy, and problems complete for all classes of the hierarchy, and give finite structures characterization for the hierarchy.

Many classes within P have been studied, including those defined by circuit complexity, or those defined on TM or RAM models by tightly bounding the polynomial time bound. Many of these latter classes suffer from the fact they are not closed under composition. One early exception to that is linear time, clearly an efficient time complexity. However, linear time is dependent on the computation model – even on the number of tapes of a multi- tape Turing machine. There have been several interesting papers recently on complexity classes near linear time. Schnorr [29], and Gurevich and Shelah [15] have considered *quasilinear* and *nearly linear* time classes (QL and NLT), classes defined on TMs and RAMs, respectively, with time bounded by $\mathcal{O}(n \log^k n)$ for some k. These classes are closed under composition, and are relatively machine-independent – Schnorr's class is invariant under the number of tapes of a multi-tape TM, while Gurevich and Shelah's class is equivalent under five different formulations of RAMs. It is an open question, whether QL=NLT. (In this paper, we will focus primarily on the Turing Machine model.)

By closing linear time under $\log n$ factors, Schnorr and Gurevich and Shelah achieve some degree of machine-independence. Another approach, taken by Grädel [11], is to consider the class of problems, TIME(n^{1+}) (relative to Turing Machines or RAMs), whose algorithms have $\mathcal{O}(n)$ as the *infimum* of their time complexities. Grädel showed that QL$\subset$TM-TIME(n^{1+}), and NLT$\subset$RAM-TIME(n^{1+}), and each of these classes display at least as much machine-independence as QL and NLT, respectively. This approach seems quite different from the approach we take, and we are not sure if there are any inclusion relations among Grädel's classes and those of the SBH, beyond the one just stated.

It can be argued that neither Turing Machines nor most RAMs accurately reflect the cost of memory access in real machines. Regan [28] has introduced a model that better approximates real-world memory access time. Within this model, he has explored linear time relative to various memory-cost measures. We have not yet considered the relationship of these classes to the SBH.

Although QL contains (as a function class) sorting, addition, and multiplication, it does not seem to contain enough functions to justify considering it as a replacement for P. Buss and Goldsmith considered extensions of this class by *sharply bounded* existential quantifiers, quantifiers of the form, $\exists y, |y| \leq log|x|$. They were able to develop a structure theory for these classes, and show that if they span P then P≠NP. Not only does it seem unlikely that these classes span P, it seems likely that there are practical problems in P that cannot be formulated using sharply bounded existential quantifiers and quasilinear-time predicates, i.e., problems that are inherently sequential.

The classes we consider are defined using sharply bounded existential *and* universal quantifiers. In some ways, this paper reflects Wrathall's work on the *linear-time hierarchy* (LTH) [31]. Like her, we define a hierarchy based on bounded quantifiers over a small time-bounded class within P, and then go on to give other characterizations of the hierarchy and the classes therein. There are several important differences, however, between the LTH and the SBH. By allowing quasilinear-time predicates, we give a certain degree of machine independence to our classes. By *sharply* bounding our quantifiers, we keep our entire hierarchy within P. There is an aesthetic cost — it seems likely that strings of the same quantifier cannot be collapsed: it is possible that $\tilde{\exists}\tilde{\exists}P_1$ has more computational power than $\tilde{\exists}P_1$, and that $\tilde{\forall}\tilde{\exists}\tilde{\forall}\tilde{\forall}P_1$ differs from $\tilde{\forall}\tilde{\forall}\tilde{\exists}\tilde{\forall}P_1$, and so on. Thus, these classes do not mirror the polynomial-time hierarchy (or the LTH) as neatly as does the LTH.

We begin by giving definitions of the relevant complexity classes. We then present a variety of problems in the SBH, define a notion of completeness, and show that some of the problems considered yield quasilinear-complete sets for classes in the hierarchy. We discuss the structure of the SBH including upward collapse theorems, and give a deceptively easy sufficient condition for P≠PSPACE. We present a finite structures (spectra) characterization of the SBH. And finally, we conclude with some open problems.

2 Definitions

Schnorr [29] introduced the classes QL and NQL, where $\mathrm{QL} = \cup_k \mathrm{DTIME}(n \log^k n)$, and $\mathrm{NQL} = \cup_k \mathrm{NTIME}(n \log^k n)$. We use the notation of Buss and Goldsmith [2] and refer to these classes as P_1 and NP_1, respectively. $P_r = \cup_k \mathrm{DTIME}(n^r \log^k n)$.

We define classes in the Sharply Bounded Hierarchy by means of *sharply bounded quantifiers* over quasilinear-time predicates. A quantifier Q is sharply bounded if it is of the form $\exists y\ |y| \leq \log|x|$ or $\forall y\ |y| \leq \log|x|$. (We write these as $\tilde{\exists}$ and $\tilde{\forall}$.) If S is a string of sharply bounded quantifiers (such as $\tilde{\forall}\tilde{\exists}\tilde{\forall}\tilde{\forall}$), then $A \in SP_1$ iff there is a P_1 relation R so that $x \in A \Leftrightarrow S\vec{y}[R(x, \vec{y}]$. Unlike the classes defined by *bounded* quantifiers, there is no evidence that iterations of a quantifier ($\tilde{\forall}\tilde{\forall}$, for example) can be collapsed to a single quantifier.

Notice that, if S is a string of r sharply bounded quantifiers, then the class $SP_1 \subseteq P_{r+1}$. We see that SBH $\subseteq$ P $\cap$ QLSPACE.

3 Examples

Buss and Goldsmith [2] have given various examples of problems in the classes $\tilde{\exists}^k$. Many of the problems considered in that paper were variants of NP-complete problems, with fixed parameters. For instance, they consider the following variant of CSAT, where a Boolean circuit C with n gates is in CSAT(k) if it has $k \log n$ inputs, and there is an input string that causes it to output a 1. We observe that there are variants of this problem in all classes of the SBH. For example, a Boolean circuit C is in $\tilde{\exists}\tilde{\forall}$CSAT if it has n gates, $2 \log n$ of which are input gates, and there is a string, x_1, of length $\log n$ such that for any string, x_2, of length $\log n$, C on input $x_1 x_2$ outputs a 1.

One can see CSAT as a variant of the P-complete problem CVP, the circuit value problem. In fact, several of the following examples are P-complete problems, as described in [12], although we also use parameter-restricted versions of problems higher in the polynomial hierarchy. Although many of our problems are P-complete, this does not seem to imply that P is contained in the SBH. In fact, we also mention a P-complete problem that seems inherently sequential, and inherently slower than P_1.

The relation, matrix multiplication is in $\tilde{\forall}^2 P_1$: given A, B, and C $n \times n$ matrices, is $A \cdot B = C$?

Perfect Matching is in $\tilde{\forall}^3 \tilde{\exists} P_1$: given a graph $G = (V, E)$, and a set of edges $M \subseteq E$, is M a perfect matching?

Unweighted not-all-equal clauses, 3SAT/FLIP [26] is in $\tilde{\forall} P_1$. If s is a truth assignment for Boolean formula B, and C is a clause of B, s gives C weight 1 if it assigns at least one T and one F to C. Given B and s, is the weight of B under s maximal over all neighbors of s?

Total Ordered Unary Generator is in $\tilde{\forall} P_1$: given an ordered context-free grammar, G, and a set T of terminal symbols, does G generate a unary string τ^* for each $\tau \in T$? (A grammar is *ordered* if, for each nonterminal A in G, each occurrence of A in the lefthand side of a production occurs before any occurrence of A on the righthand side of a production.) In [2], it is shown that the set of ordered unary generators is in (and is complete for) $\tilde{\exists} P_1$. This variant follows easily from their proof of membership.

However, there are certain P-complete problems that do not appear to be in the SBH. One example is the problem of Unit Resolution. That is clearly in P_2, and not apparently in P_1, even with the help of sharply bounded quantifiers.

4 Complete Sets

The notion of quasilinear-time computation can also be applied to transducer Turing machines. Since all of the classes in the SBH are closed under quasilinear-time functions, it is natural to define a notion of quasilinear-time, many-one complete sets for these classes. This notion has been explored in [29, 15, 2]. Schnorr showed, in [29], that SAT is $\leq^{ql}$-complete for nondeterministic quasilinear time (NQL, a.k.a. NP_1). We can show, using the same techniques, that

Theorem 1 *QBF, the set of true quantified Boolean formulas, is $\leq^{ql}$-hard for any class in the SBH.*

Buss and Goldsmith gave various problems that are complete for classes of the form $\tilde{\exists}^k P_1$. Variants of some these can be shown to be $\leq^{ql}$- complete for all classes in the SBH hierarchy, for instance,

- There are examples of Sharply-Bounded Quantified Circuit Satisfiability in each level of the SBH, where an instance of SBQCS is "$\tilde{\exists} x_1 \tilde{\forall} x_2$ such that circuit C on input $x_1 x_2$ outputs a 1."
- There are examples of Sharply-Bounded Shortsat at each level of the SBH, where and instance of SBSS is, "$\tilde{\forall} x_1 \tilde{\forall} x_2 \tilde{\exists} x_3$ such that $x_1 x_2 x_3$ is an initial (partial) truth assignment to Boolean formula B that causes B to unravel.[1]

5 The Structure of the Classes

Like most hierarchies of complexity classes, the SBH exhibits some forms of upward collapse/downward separation. Some of these collapses concern the compression of like quantifiers (from $\tilde{\exists}\tilde{\exists}$ to $\tilde{\exists}$, for instance), as well as (or instead of) the quantifier-swapping we expect of collapsing results.

Lemma 1 *If C is a class in the SBH, and $\tilde{\exists}^{k+1} C \subseteq \tilde{\exists}^k C$, then for all $m > k$, $\tilde{\exists}^m C \subseteq \tilde{\exists}^k C$.*

This is proved (as in [2]) by a straightforward padding argument.

Lemma 2 *If C is a class in the SBH, and $k > k'$, and if $\tilde{\exists}^k C = \tilde{\forall}^{k'} C$, then for all $m > k$, $\tilde{\exists}^m C \subseteq \tilde{\exists}^k C$.*

[1] The notion of *unraveling* is more commonly expressed as, "this initial partial assignment *forces* a satisfying assignment for B," where an assignment of $v_1 = T$ forces $v_2 = T$ in the clause $(\overline{v}_1 \vee v_2)$.

Proof: If $\tilde{\exists}^k C = \tilde{\forall}^{k'} C$, then $\tilde{\exists}^{k'} C = \tilde{\forall}^k C$, so $\tilde{\exists}^k C = \tilde{\exists}^{k'}$, and in particular, $\tilde{\exists}^k C \subseteq \tilde{\exists}^{k'} C$. So $\tilde{\exists}^{k+1} C \subseteq \tilde{\exists}^k C$, and the result follows by Lemma 1. ∎

However, we know of no interesting consequences if $\tilde{\exists}^k C = \tilde{\forall}^k C$, since strings of like quantifiers don't appear to collapse. Thus, our intuition, developed by working with (bounded) quantifiers, or in Boolean circuits, does not help.

The following lemma was proved by Schnorr in [29], using results of Pippenger [27].

Lemma 3 *SAT is $\leq^{ql}$-complete for P_1.*

Theorem 2 *If P⊆SBH, then P≠PSPACE.*

Proof: The proof of theorem 2 follows immediately from the following observation: if P=PSPACE, then QBF is contained in DTIME($\mathcal{Q}(n^k)$) for some k, so SBH is as well, by Lemma 3. Thus, SBH⊂P. ∎

This has a surprising corollary in the following theorem.

Theorem 3 *If $P_2 \subseteq$ SBH, then P≠PSPACE.*

Proof:

Suppose that $P_2 \subseteq SP_1$, for some string, S, of sharply bounded quantifers.

Lemma 4 *If $P_2 \subseteq SP_1$ for some string S of sharply-bounded quantifiers, then $P_4 \subseteq SSP_1$.*

Let $A \in P_4$, and let $B = \{y : \ y = x10^{|x|^2 - |x| - 1} \wedge x \in A\}$. Then $B \in P_2$, and thus in SP_1. To decide $A(x)$, in time $\mathcal{Q}(n^2)$, write $y = x10^{|x|^2 - |x| - 1}$, and decide (in time $\mathcal{Q}(|x|^2)$, with quantifiers S), if $y \in B$. In other words, $A \in SP_2 \subseteq SSP_1$.

Corollary 1 *If $P_2 \subseteq SP_1$ for some string S of sharply-bounded quantifiers, then $P \subseteq$SBH.*

This follows by induction from Lemma 4. Thus, by Theorem 2, P≠PSPACE. ∎

6 A Finite Structures Characterization of SBH

The relation class SBH can be described as the set of relations $R(\vec{x})$ such that, for some relation $T(\vec{x},\vec{y}) \in P_1$ and some

$$R(\vec{x}) \iff (\exists y_1 \leq n)(\forall y_2 \leq n) \cdots T(\vec{x},\vec{y}).$$

where $n = \max(|\vec{x}|)$. The same class can equivalently be defined in a finite-models manner like Immerman's [19] machine-independent characterization of the circuit class, uniform AC^0, as extended by Clote. [5]

Definition 1 *A j-ary relation $R(\vec{x})$ is said to be* first-order definable in P_1 *(in short, $R \in FO(P_1)$) if there are relations $R_1, R_2, \ldots R_m \in P_1$ of arities $j+k_1, j+k_2, \ldots j+k_m$ respectively, a first-order sentence θ containing no function or constant symbols, and relation symbols $S_1, S_2, \ldots S_m$ of arities $k_1, k_2, \ldots k_m$ respectively, such that*

$$R(\vec{x}) \iff \langle \{0,1,\ldots \max(|\vec{x}|)\}, \lambda\vec{y}.R_1(\vec{x},\vec{y}), \lambda\vec{y}.R_2(\vec{x},\vec{y}), \ldots \lambda\vec{y}.R_m(\vec{x},\vec{y})\rangle \models \theta.$$

Theorem 4 $SBH = FO(P_1)$

Proof: Suppose $R \in SBH$, i.e. for some quantifiers $Q_1, Q_2, \ldots Q_r \in \{\exists, \forall\}$

$$R(\vec{x}) \iff \mathbf{N} \models (Q_1 y_1 \leq n) \cdots (Q_r y_r \leq n) T(\vec{x},\vec{y}).$$

Since T contains no function or constant symbols, a particular variable assignment with values $\leq n$ satisfies T in $\mathbf{N}$ iff it satisfies T in the finite structure $\{0,1,\ldots n\}$. Now replace each occurrence of a relation symbol $R_i(\vec{x},\vec{y})$ in T by the symbol $S_i(\vec{x})$, interpret S_i by $\lambda\vec{y}.R_i(\vec{x},\vec{y})$, and the resulting formula is an $FO(P_1)$ definition of R.

Conversely, suppose $R \in FO(P_1)$, i.e. there are relations $R_1, R_2, \ldots R_m$ and a first-order sentence θ as above such that

$$R(\vec{x}) \iff \langle \{0,1,\ldots \max(|\vec{x}|)\}, \lambda\vec{y}.R_1(\vec{x},\vec{y}), \lambda\vec{y}.R_2(\vec{x},\vec{y}), \ldots \lambda\vec{y}.R_m(\vec{x},\vec{y})\rangle \models \theta.$$

Replace each relation symbol $S_i(\vec{y})$ in θ by $R_i(\vec{x},\vec{y})$, calling the result θ'. Put θ' into prenex form, and call its quantifier-free matrix $T(\vec{x},\vec{y})$. As before, since T contains no function or constant symbols, it is true of particular variables $\vec{x},\vec{y}$ in $\{0,1,\ldots n\}$ iff it is true in $\mathbf{N}$, so

$$R(\vec{x}) \iff \mathbf{N} \models (Q_1 y_1 \leq n)(Q_2 y_2 \leq n) \cdots T(\vec{x},\vec{y}),$$

where $Q_1, Q_2, \ldots Q_r$ are the quantifiers in the prenex form of θ'. Thus $R \in SBH$. ∎

This technique can be generalized, both by changing the base relation class P_1 and by expanding the language of bounding terms allowable in bounded quantifiers. In particular, if the base relation class P_1 is changed to the finite set $\{\leq, BIT\}$ and our attention is restricted to unary relations $R(x)$, the resulting class is exactly Immerman's class FO (= uniform AC^0).

7 Open Questions

One of the most challenging questions raised by this work is the question of whether P_2, or any deterministic time class sufficiently above P_1, is contained in the SBH, since this would imply that P≠PSPACE. Barring that, we would like to know the relationship of the SBH to nondeterministic quasilinear time. For that matter, we would like to know if deterministic nearly linear time (QL on RAM machines), as defined and studied by Gurevich and Shelah [15], is contained in the SBH.

It would be interesting to consider the classes defined by polylog many sharply bounded quantifiers. These classes would be analogous to the limited nondeterminism considered by Kintala and Fischer [22] and by Álvarez, Díaz and Torán [1]. They considered classes between P and NP; our classes would be between P and PSPACE, and might perhaps yield some insight into the P=?PSPACE question.

Acknowledgements

The authors would like to thank Peter Clote for suggesting that they work on this material, and making several helpful suggestions, and Michael Saks for pointing out that $\tilde{\forall} P_1$ contained interesting problems.

References

[1] C. Álvarez, J. Díaz and J. Torán, "complexity classes with complete problems between P and NP-complete," in *Foundations of Computation Theory,* Lecture Notes in Computer Science 380, Springer-Verlag, 1989, pp. 13-24.

[2] J. F. Buss and J. Goldsmith, "Nondeterminism within P," *SIAM J. Comput.* (to appear, 1993).

[3] S. R. Buss, **Bounded Arithmetic,** Bibliopolis, Napoli 1986.

[4] A. Chandra, D. Kozen, and L. Stockmeyer, "Alternation," *J. of the ACM* **28** (1981), 114-133.

[5] P. Clote, sequential, machine-independent characterizations of the parallel complexity classes ALOGTIME, AC^k, NC^k, and NC." **Feasible Mathematics,** S. Buss and P. Scott, Eds., Perspectives in Computer Science, Birkhauser-Boston, New York (1990).

[6] A. Cobham, "The intrinsic computational complexity of functions," *Proc. 1964 International Congress for Logic, Methodology, and Philosophy of Sciences,* Y. Bar-Hillel, Ed., North Holland (1965), 78–87.

[7] A. Condon, "Strict P-completeness," *manuscript* (1992).

[8] S. A. Cook, "Computational complexity of higher type functions," *Proc. ICM* (1990).

[9] J. Edmonds, "Paths, trees, flowers," *Canadian J. Math.* **17** (1965), 449–67.

[10] J. Girard, A. Scedrov, and P. Scott, "Bounded linear logic: a modular aprroach to polynomial time computability (extended abstract)," **Feasible Mathematics,** S. Buss and P. Scott, Eds., Perspectives in Computer Science, Birkhauser-Boston, New York (1990), 195–207.

[11] E. Grädel, "On the notion of linear time computability," *International J. Foundations of Comp. Sci.* **1** (1990), 295–307.

[12] R. Greenlaw, H. J. Hoover, and W. L. Ruzzo, "A compendium of problems complete for P (Preliminary version)," *Dept of Computer Science and Engineering, University of Washington, Technical Report TR 91-05-01* (1991).

[13] Y. Gurevich, "Algebras of feasible functions," *Proc. 24th FOCS,* IEEE Computer Society Press (1983), 210–214.

[14] Y. Gurevich, "Logic and the challenge of computer science," in **Current Trends in Theoretical Computer Science,** E. Börger, Ed., Computer Science Press (1987).

[15] Y. Gurevich and S. Shelah, "Nearly linear time," in A. R. Meyer and M. A. Taitslin, eds., *Logic at Botik '89,* Lecture Notes in Computer Science 363, Springer-Verlag, 1989, pp. 108–118.

[16] Y. Gurevich and S. Shelah, "Fixed-point extensions of first-order logic," *Annals of Pure and Applied Logic* **32** (1986), 265–280.

[17] N. Immerman, "Relational queries computable in polynomial time," *Info. and Control* **68** (1986), 86–104.

[18] N. Immerman, "Languages that capture complexity classes," *SIAM J. Comp* **16** (1987), 760–778.

[19] N. Immerman, "Expressibility and parallel complexity," *SIAM J. Comp* **18** (1989).

[20] R. M. Karp, "Reducibility among combinatorial problems," *Complexity of Computer Computations,* R.E. Miller and J.W. Thatcher, Eds., Plenum Press, New York (1972), 85–104.

[21] R. Kaye, "Characterizing some low complexity classes using theories of arithmetic," M.Sc. thesis, University of Manchester, 1985.

[22] C. M. R. Kintala and P. C. Fischer, "Refining nondeterminism in relativized polynomial-time bounded computations," *SIAM J. Comput.* 9 (1980) 46–53.

[23] D. Leivant, "A foundational delineation of computational feasibility," *Proc. IEEE Logic in Comp. Sci.* (1991).

[24] D. Leivant, "Inductive definitions over finite structures," *Info. and Comp.* **89** (1990), 95–108.

[25] C. H. Papapdimitriou, "A note on the expressive power of PROLOG," *Bull. of the EATCS* **26** (June 1985), 21–23.

[26] C. H. Papadimitriou, A. A. Schäeffer, and M. Yannakakis, "On the complexity of local search," *Proc. 22nd STOC* (1990), 438–445.

[27] N. Pippenger, "Fast simulation of combinational logic circuits by machines without random-access storage," in *Proc. 15th Allerton Conference on Communication, Control, and Computing,* 1977, pp. 25–33.

[28] K. Regan, personal communication.

[29] C. P. Schnorr, "Satisfiability is quasilinear complete in NQL," *J. Assoc. Comput. Mach.* 25 (1978) 136–145.

[30] M. Vardi, "Complexity and relational query languages," *Proc. 14th STOC* (1982), 137–146.

[31] C. Wrathall, "Rudimentary predicates and the linear time hierarchy," *SIAM J. of Comput.* **7** (1978), 194–209.