On Superpolylogarithmic Subexponential Functions*

Alan T. Sherman*
Computer Science Department
University of Maryland Baltimore County
Baltimore, Maryland 21228
and
Institute for Advanced Computer Studies
University of Maryland College Park
College Park, Maryland 20742

June 21, 1990
(revised April 1, 1991)

Abstract

A *superpolylogarithmic subexponential function* is any function that asymptotically grows faster than any polynomial of any logarithm but slower than any exponential. We present a recently discovered nineteenth-century manuscript about these functions, which in part because of their applications in cryptology, have received considerable attention in contemporary computer science research. Attributed to the little-known yet highly-suspect composer/mathematician Maria Poopings, the manuscript can be sung to the tune of "Supercalifragilisticexpialidocious" from the musical *Mary Poppins*. In addition, we prove three ridiculous facts about superpolylogarithmic subexponential functions. Using novel extensions to the popular DTIME notation from complexity theory, we also define the complexity class SuperPolyLog/SubExp, which consists of all languages that can be accepted within deterministic superpolylogarithmic subexponential time. We show that this class is notationally intractable in the sense that it cannot be conveniently described using existing terminology. Surprisingly, there is some scientific value in our notational novelties; moreover, students may find this paper helpful in learning about growth rates, asymptotic notations, cryptology, and reversible computation.

Keywords

Algorithms, asymptotic notation, complexity theory, cryptography, cryptology, DTIME, mathematical humor, Maria Poopings, Mary Poppins, musical computer science, reversible computation, Supercalifragilisticexpialidocious, superpolylogarithmic subexponential functions, SuperPolyLog/SubExp.

*A preliminary version of this paper appeared in two parts in *SIGACT News*, **22**:1 (winter 1991), Whole Number 78, 65–73, and **22**:2 (spring 1991), Whole Number 79, 51–56.

1 Introduction

In an abandoned wine cellar beneath the Ratskeller in Wein-am-Rhein, construction workers recently discovered an intriguing manuscript. This manuscript, dated 1824, is the earliest known reference to the important class of functions that asymptotically grow faster than any polynomial of any logarithm but slower than any exponential—the *superpolylogarithmic subexponential functions*.[1] Although difficult to decipher through numerous beer stains, the manuscript appears to bear the inscription "M.P.", which I believe stands for Maria Poopings, who is believed to be the little-known illegitimate daughter of the dubious German composer P. D. Q. Bach (1807–1742)? [32] and the brilliant French mathematician Marie-Sophie Germain (1776–1831) [13]. Reprinted in Section 3, the text of this manuscript fits remarkably well—especially after an evening at the Ratskeller—the tune "Supercalifragilisticexpialidocious" from the 1964 Walt Disney musical *Mary Poppins* [28, 42, 43].[2] The next two paragraphs explain my mathematical and personal motivations for studying Poopings's work.

Since Poopings's pioneering paper, numerous mathematicians and computer scientists have independently used and studied superpolylogarithmic subexponential functions in a variety of applications—seriously! For example, in 1979 Adleman [1] presented a superpolylogarithmic subexponential-time algorithm for solving the discrete logarithm problem, a problem which has formed the basis for some cryptographic schemes. Similarly, in 1985, H. W. Lenstra [19, 20, 4, 36] designed a new superpolylogarithmic subexponential-time algorithm based on elliptic curves for splitting integers.[3] Since the RSA cryptosystem [30]—the most widely used public-key cryptosystem today—can be broken by factoring the modulus, the complexity of integer factoring is of great interest in cryptology. Given any composite integer n, Lenstra's algorithm extracts a nontrivial divisor p of n; under some heuristic assumptions, Lenstra proves that his algorithm runs in expected time $O((\ln n)^2 e^{\sqrt{(2+o(1))\ln p \ln\ln p}})$.[4] Although other methods seem to work better in practice [29], to date, Lenstra's splitting algorithm yields the asymptotically fastest general-purpose (albeit based in part on heuristic assumptions) integer factoring algorithm published in the open literature.[5]

Although the many modern references to superpolylogarithmic subexponential functions sufficiently justify their further study, what really motivated this paper is the following editorial experience. In my recent paper with Levine [22] on reversible computation, I used the word "superlogarithmically" to express the fact that a certain function grows faster than any logarithm. In this context the following editorial question arose: should the word "superlogarithmically" be hyphenated? My initial preference was to hyphenate

[1]For a precise definition of a superpolylogarithmic subexponential function, see Appendix B.

[2]The words and music to the tune "Supercalifragilisticexpialidocious" were composed by Richard M. Sherman and Robert B. Sherman, who to the best of my knowledge, have no relationship to my father Richard B. Sherman.

[3]To *split* an integer n is to find any nontrivial divisor of n; to *factor* n is to find all prime factors of n. Although any splitting algorithm can be transformed into a factoring algorithm, the distinction between between splitting and factoring is especially important when the running time of the splitting algorithm depends on the divisor found, as happens with Lenstra's algorithm.

[4]Throughout this paper, let $\ln = \log_e$ and $\lg = \log_2$.

[5]For a slightly faster special-purpose algorithm that works only for a certain subset of integers, see the number field sieve described by A. K. Lenstra, H. W. Lenstra, Masasse, and Pollard [21, 17], which I shall refer to as *Lenstras'* algorithm.

"super-logarithmically" because "superlogarithmically" is a mouthful. Our editor, however, preferred not to hyphenate this word. I now agree: the *American Heritage Dictionary* [25, p. 1290] explains that words beginning with the prefix "super" are not usually hyphenated, unless the stem is capitalized (*e.g.* "supernumerary" but "super-American"). However, *Webster's Ninth New Collegiate Dictionary* [35, p. 1183] lists "super-duper," and Fowler [11, p. 604] hyphenates "super-priority" and "super-cinema."[6] Since my editor did not consider "superlogarithmically" sufficiently mouthfilling, I decided to create—I cannot resist—a supermouthful.[7] This experience with hyphens motivated me to write about superpolylogarithmic subexponential functions.

The rest of this paper is organized in three sections followed by three appendages. Section 2 proves three ridiculous facts about superpolylogarithmic subexponential functions. Section 3 reprints the text of Maria Poopings's 1824 manuscript. Section 4 offers some concluding observations, reviews the literature on musical computer science, and suggests avenues for future research. The "Codetta" describes my search for Maria Poopings's mother—a frivolous folly into the history of European woman mathematicians. Appendix A gives detailed performance notes on how to sing Poopings's manuscript. Tastefully segregated from the body of the paper, the serious Appendix B explains what superpolylogarithmic subexponential functions are in a precise mathematical way. This appendix helps clarify confusion caused by the common practice of many authors to use the term "subexponential" without any precise definition in mind. Appendix B also defines several complexity classes related to superpolylogarithmic subexponential functions. To define these classes in a convenient way, Appendix B proposes and exploits expressive extensions to the the DTIME notation from complexity theory. The impatient reader may like to skip directly to Section 3.

2 Results

Containing a total of 34 letters (including 18 distinct letters), the expression "superpolylogarithmic subexponential" has a rich distribution of letters. This distribution, which covers approximately 69% of the English alphabet, forms the basis of the three results presented in this section.

The first result is that the phrase "superpolylogarithmic subexponential" contains 11 repeated letters, including the letters "e", "i", "l", "o", and "p" each of which appear exactly 3 times. Although this result can be proven constructively, it is noteworthy that the existence of at least one repeated letter can be proven nonconstructively using the Pigeonhole Principle: the phrase contains 34 letters, but there are only 26 letters in the English alphabet.

[6] Fowler prefers to restrict the use of the prefix "super" to its primary sense of "above" or "transcending" (*e.g.* "superstructure"). As for the common usage of the prefix "super" to mean "of a superior kind" (*e.g.* "super-different"), Fowler [11, p. 604] laments and admonishes, "[This usage] … is so evidently convenient that it is vain to protest when others indulge in it, and so evidently catachrestic that it is worth while to circumvent it oneself when one can do so without becoming unintelligible." I presume Fowler would have approved of "superpolylogarithmic" but not of "supercomputer."

[7] Paradoxically, the editor initially wanted to hyphenate the word "trade-off" despite my preference to use the shorter, simpler unhyphenated "tradeoff" (also preferred by the 1976 New College edition of the American Heritage Dictionary [25, p. 1360]), and despite my objection to the proliferation of hyphens in the phrase "time-space trade-off."

Stated as Theorem 1 and Corollary 1, the next two results deal with complexity theory and with the nature of mathematics; each is proven by anagramming. Theorem 1 establishes a lower bound on the time complexity of Mary Poppins's so-called 9-count algorithm—whatever that algorithm may be. Corollary 1 proves, up to an additive constant, a postulate that many mathematicians have long accepted without proof.

Theorem 1.
Mary Poppins's 9-count algorithm $\in$ *superpolylogarithmic subexponential*(m).

Proof (by anagramming).
Using Roman's identity $9 = \mathrm{IX}$, it is possible to rearrange the letters of the phrase "Mary Poppins's 9-count algorithm" into letters of the phrase "superpolylogarithmic subexponential (m)," leaving a remainder of "bull + e^3." □

Corollary 1.
Up to the additive constant c, it is true that Math Science is Pure Glory.

Proof.
Starting with the phrase "Mary Poppins's 9-count algorithm," we add the remainder term from Theorem 1 to produce

$$\text{Mary Poppins's 9-count algorithm} + (\text{bull} + e^3). \tag{1}$$

Next, using the facts that $9 = \mathrm{IX}$ and "pi" $= \pi$, the letters of Expression 1 can be rearranged to obtain

$$\text{Math Scien\_e is Pure Glory} + (L - L)a_o b_o \pi \mathrm{P}[x \geq \frac{tu}{nm}]. \tag{2}$$

Finally, eliminating the 0-valued summand from Expression 2 yields the desired aphorism, up to the missing letter "c". □

In addition to these three results, the derivative adverb "superpolylogarithmically," with exactly 24 letters, is to the best of my knowledge the longest unhyphenated nonmedical technical word ever used in any paper on computer science (the first use being in this paper).[8]

3 Lyrics

The following lyrics are reprinted from the recently discovered 1824 manuscript attributed to Maria Poopings.

[8]The qualification "technical" is important here because this paper also uses the nontechnical word "Supercalifragilisticexpialidocious," which has exactly 34 letters; the qualification "nonmedical" is included for conservatism (see [24, pp. 145–146]).

Superpolylogarithmic Subexponential Functions

(Sung to the tune of "Supercalifragilisticexpialidocious" [28, 42, 43].)[9]

Um diddle diddle diddle, um diddle ay!
Um diddle diddle diddle, um diddle ay!

CHORUS:
Superpolylogarithmic subexponential functions!
Faster than a polylog but slower than exponential.
Even though they're hard to say, they're truly quintessential.
Superpolylogarithmic subexponential functions!

Um diddle diddle diddle, um diddle ay!
Um diddle diddle diddle, um diddle ay!

For Alice to send a message through to Bob when Eve's eavesdropping,
do use a trapdoor one-way function—not a one-key mapping.
First take a message x, let's say, and raise it to the e;
then mod it out by p times q but keep these secretly. Oh!

(CHORUS)

Um diddle diddle diddle, um diddle ay!
Um diddle diddle diddle, um diddle ay!

The process takes but poly-time and appears to be secure:
why even just a single bit is one over polylog pure.
Though Alice thinks that Eve must spend time at least exponential,
by using Lenstra's elliptic curves, Eve splits n subexponentially. Oh!

(CHORUS)

Um diddle diddle diddle, um diddle ay!
Um diddle diddle diddle, um diddle ay!

Most computations dissipate a lot of energy;
we remove the heat with water but there's a better strategy.
Since thermodynamics does not apply when info is not doomed,
the laws of physics don't require that power be consumed. Oh!

(CHORUS)

[9] *Italicized words* should be sung; unitalicized words should be spoken. For help with inside jokes and references, see [3, 6, 7, 8, 19, 20, 30, 36].

Um diddle diddle diddle, um diddle ay!
Um diddle diddle diddle, um diddle ay!

Now Bennett said in '73, to run a program P,
you simulate the program P, but do so reversibly.
The problem with this method is that space is exponential,
so trade off time to save on space—this really is essential! Oh!

(CHORUS)

Um diddle diddle diddle, um diddle ay!
Um diddle diddle diddle, um diddle ay!

Did you know if you invert one, you get a
funtionential subexporithmic logapolyrepus?
But that's quite a singularity! *So,*

If you are in an oral exam and cannot find the way,
just summon up these words and then you've got a lot to say.
But better use them carefully or you could fail the test.
A professor once asked me,
"What do you call functions that grow faster than any polylogarithm but slower than exponential?"
There're,

Superpolylogarithmic subexponential functions!
Superpolylogarithmic subexponential functions!
Superpolylogarithmic subexponential functions!
Superpolylogarithmic subexponential functions!

4 Conclusion

The discovery of Poopings's manuscript helps fill a significant gap in the body of scientific knowledge: not enough work has been done in the area of *musical computer science*, the study of computer science through music. Although much research has been carried out on computer music (see Roads and Strawn [31], for example), and although former Harvard mathematics Teaching Fellow Tom Lehrer has recorded and published music on new math [18, 41] and on the Russian mathematician Nicolai Ivanovich Lobachevsky (1793–1856) [18, 40],[10] I know of relatively little published music on computer science. The only refereed such publications I am aware of are Knuth's [16] seminal paper on the complexity of songs and Steele's "The Telnet Song" [34]. As for readily-available recorded music, I recommend the collection "The funniest computer science songs," produced by Vince Emery [38]; additional individual songs are buried

[10] In conflict with information listed by Tom Lehrer about Lobachevsky, some sources (*e.g. The New Encyclopædia Britannica* [23]) give the alternate transliteration and birth date "Nicolay" and 1792, respectively.

in various tapes sold by the science fiction and fantasy art and music distributor Firebird Arts and Music, Inc.[11] Unfortunately, most work on musical computer science is communicated only through obscure off-key performances, unpublished and unpublishable manuscripts, and little-known computer bulletin boards (*e.g.* rec.music.dementia).[12] Part of the problem is that of the $151,393,000 spent by the National Science Foundation (NSF) in 1989 on program grants and contracts by the Directorate for Computer and Information Science and Engineering, not one dollar was invested in musical computer science [12]. By contrast, according to Tom Lehrer [37], "The sales to date of Mr. Lehrer's records have been phenomenal—over forty copies in the United States alone...." Perhaps NSF should undertake an initiative on musical computer science and use profits from the resulting work to supplement its computer science research budget.

This paper raises several open problems. First, it would be fascinating to know if there are any additional manuscripts by Poopings waiting to be discovered. Researchers eager to pursue this problem may find it helpful to study the work of Tom Lehrer and of Allan Sherman [39].[13] Second, and less seriously, it would be interesting to explore the relative power of nondeterminism versus determinism for Turing machines that run in superpolylogarithmic subexponential time. To this end, as defined in Appendix B, let *SuperPolyLog/NSubExp* denote the class of languages that can be accepted within deterministic superpolylogarithmic subexponential time, and let *NSuperPolyLog/NSubExp* denote the class of languages that can be accepted within nondeterministic superpolylogarithmic subexponential time. Although SuperPolyLog/SubExp is contained in NSuperPolyLog/NSubExp, it is unknown if this containment is proper. Third, there are probably more facts that can be established along the lines of Theorem 1.

I would like to close with a personal recommendation—watch the movie *Mary Poppins*. Winner of six Academy Awards (Best Actress: Julie Andrew, Best Film Editing, Best Song: "Chim Chim Cher-ee," Best Score, Best Visual Effects, and a special technical award for combining live action and animation) [43], it is an enjoyable musical for viewers of all ages. When listening to the song "Supercalifragilisticexpialidocious," remember the manuscript of Maria Poopings.

Codetta: How I Found Maria Poopings's Mother

In tracing Maria Poopings's ancestry, I focused my search on famous female European mathematicians who were contemporaries of P. D. Q. Bach. This genealogical endeavor sadly brought to my attention a lack of awareness among many current mathematicians of math history and of the contributions of woman mathematicians. Whenever I met a mathematician I would ask: "Who was the most famous female European mathematician of the late eighteenth century?" Infallibly the answer would be silence. Loosening my constraints, I would follow up: "Can you name any famous female European mathematician?" Again, there would be silence. In

[11] Mailing address: Firebird Arts and Music, Inc., P.O. Box 453, El Cerrito, CA 94530.

[12] William Gasarch, who describes himself as "a reasonably knowledgeable collector of musical computer science," has $\Theta(25)$ songs about computer science, of which $\Theta(17)$ are readily available.

[13] To the best of my knowledge, there is no relationship between Allan Sherman and myself beyond the similarity of our names. Although Allan Sherman never wrote any computer science songs, I recommend his work here because I like his humorous ballads; besides, doing so gives me an opportunity to point out the amusing similarity of our names.

desperation I would continue: "Can you name *any* female mathematician?" Although two people mentioned Augusta Ada Lovelace (1815–1852), for whom the programming language Ada was named, or Grace Brewster Murray Hopper (1906–), the mother of COBOL, and although one computer scientist mentioned the modern Dutch logician A. S. Troelstra, everyone else—approximately ten people from my unscientific sample—drew a blank.

Relentless to discover the identity of Maria Poopings's mother, I went to the library and looked through numerous books on the history of mathematics. In vain I browsed Bell's *Men of Mathematics* [5], Muir's *Of Men and Numbers: The Story of the Great Mathematicians* [26], and other chauvinistic works. Then I struck gold with Grinstein and Campell's *Women of Mathematics: A Bibliographic Sourcebook* [14], which describes over forty female mathematicians, including Hypatia (370?–415), Maria Gaetana Agnesi (1718–1799), Marie-Sophie Germain (1776–1831), Mary Fairfax Greig Somerville (1780–1872), Sofia Vasilevna Kovalevskaia (1850–1891), and Emmy Noether (1882–1935).[14]

Sophie Germain was a remarkable person. Born in Paris into the family of a prosperous silk merchant, Germain became interested in mathematics in part through reading one of her father's books on the history of mathematics [13]. Discouraged by her parents and unable to attend school because she was female, Germain assiduously studied mathematics on her own. Fortunately, she was able to correspond with Lagrange, Legendre, and Gauss, but fearing sexual discrimination, she did so under the pseudonym "M. LeBlanc." To their credit, Lagrange and Gauss did not lose their high respect for her work after learning the true identity of M. LeBlanc. According to Gray [13, p. 50], however, Siméon Denis Poisson "[appropriated some] of her work without attribution." Although much of her work lacks rigor—partly a reflection of her lack of formal training—Germain made significant contributions in number theory, elasticity, and philosophy, including proving a special case of Fermat's Last Theorem. Gray [13, p. 50] acclaims, "There is no doubt that her research is the most significant done by a woman mathematician before Sofia Kovalevskaia." Her accomplishments are especially impressive given that she was denied the education, easy access to professionals, and critical feedback enjoyed by male mathematicians of her time.

Acknowledgments

I would like to thank William Gasarch and Clyde Kruskal for useful editorial comments. In addition, I am grateful to Gasarch for introducing me to Vince Emery's "The funniest computer science songs" [38] and Firebird Arts and Music, Inc. I am also grateful to Katherine Kruskal for sight-singing an early version of Poopings's manuscript and to Thomas Cain for helping me create one of the rhymes. Finally, I would like to thank Charles Leiserson for pointing out the embedding coMpUter ScIenCe.

[14] For additional references on women in mathematics, see Dauben [10, pp. 428–434].

References

[1] Adleman, Leonard, "A subexponential algorithm for the discrete logarithm problem with applications to cryptography (abstract)" in *Proceedings of the 20th Annual Symposium on Foundations of Computer Science*, IEEE Computer Society 79 CH1471-2C (1979), 55–60.

[2] Aho, Alred V.; John E. Hopcroft; and Jeffrey D. Ullman, *Data Structures and Algorithms*, Addison-Wesley (Reading, MA, 1983).

[3] Alexi, Werner; Benny Chor; Oded Goldreich; and Claus P. Schnoor, "RSA and Rabin functions: Certain parts are as hard as the whole," *SIAM Journal on Computing*, **17**:2 (April 1988), 194–209.

[4] Bach, Eric, "Lenstra's algorithm for factoring with elliptic curves (expose)," Computer Science Department, University of Wisconsin, Madison (February 27, 1985), unpublished manuscript.

[5] Bell, E. T., *Men of Mathematics*, Simon and Schuster (New York, 1937).

[6] Bennett, Charles H., "Time/space trade-offs for reversible computation," *SIAM Journal on Computing*, **18**:4 (August 1989), 766–776.

[7] Bennett, Charles H., "Logical reversibility of computation," *IBM Journal of Research and Development*, **17**:6 (November 1973), 525–532.

[8] Bennett, Charles H.; and Rolf Landauer, "The fundamental physical limits of computation," *Scientific American*, **253**:1 (July 1985), 48–56.

[9] Cormen, Thomas H.; Charles E. Leiserson; and Ronald L. Rivest, *Introduction to Algorithms*, MIT Press and McGraw-Hill (Cambridge, MA, and New York, 1990).

[10] Dauben, Joseph W., *The History of Mathematics from Antiquity to the Present: A Selective Bibliography*, Garland (New York, 1985).

[11] Fowler, H. W., *A Dictionary of Modern English Usage*, second edition revised by Sir Ernest Gowers, Oxford University Press (New York, 1985).

[12] "FY 1989 awards by state and institution," National Science Foundation, Washington, D.C.

[13] Gray, Mary W., "Sophie Germain (1776–1831)" in [14], 47–56.

[14] Grinstein, Louise S.; and Paul J. Campbell, eds., *Women of Mathematics: A Bibliographic Sourcebook*, Greenwood Press (New York, 1987).

[15] Hopcroft, John E.; and Jeffrey D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley (Reading, MA, 1979).

[16] Knuth, Donald E., "The complexity of songs," *CACM*, **27**:4 (April 1984), 344–346.

[17] Kolata, Gina, "Giant leap in mathematics: 155 divided to 0," *The New York Times* (June 20, 1990), A12.

[18] Lehrer, Tom; and Frank Metis, "Too many songs by Tom Lehrer (with not enough drawings by Ronald Searle)," Pantheon Books (New York, 1981). [Music scores arranged for voice, piano, and guitar, including all songs from the musical review *Tomfoolery* (with five drawings by Ronald Searle)]

[19] Lenstra, A. K. Jr., "Factoring integers with elliptic curves," *Annals of Mathematics*, **126**:3 (1987), 649–673.

[20] Lenstra, A. K.; and H. W. Lenstra Jr., "Algorithms in number theory" in *Handbook of Theoretical Computer Science*, A. Meyer, M. Nivat, M. Patterson, and D. Perrin, eds., North-Hollad (Amsterdam), to appear.

[21] Lenstra, A. K.; H. W. Lenstra; M. S. Manasse; and J. M. Pollard, "The number field sieve" in *Proceedings of the 22nd Annual Symposium on Theory of Computing*, ACM Press (1990), 564–572.

[22] Levine, Robert Y.; and Alan T. Sherman, "A note on Bennett's time-space tradeoff for reversible computation," *SIAM Journal on Computing*, **19**:4 (August 1990), 673–677.

[23] "Lobachevsky, Nicolay Ivanovich" in *The New Encyclopædia Britannica*, 15th edition, vol. 7, Encyclopædia Britannica, Inc. (Chicago, 1989), 428–429.

[24] McFarlan, Donald; Norris D. McWhirter, Michelle D. McCarthy, and Mark Young, eds., *Guinnesss Book of Records 1991*, Facts On File (New York, 1990).

[25] Morris, William, ed., *The American Heritage Dictionary of the English Language*, New College edition, Houghton Mifflin (Boston 1976).

[26] Muir, Jane, *Of Men and Numbers: The Story of the Great Mathematicians*, Dood, Mead, & Co. (New York, 1961).

[27] Munkres, James R., *Topology: A First Course*, Prentice-Hall (Englewood Cliffs, NJ, 1975).

[28] Okun, Milton, ed., *Vocal Selections from Walt Disney's Marry Poppins* (music and lyrics by Richard M. Sherman and Robert B. Sherman), Hal Leonard Publishing Corporation (Winona, MN, 1984). [Music scores arranged for voice, piano, and guitar]

[29] Pomerance, Carl, "The quadratic sieve factoring algorithm," *Advances in Cryptology: Proceedings of Eurocrypt 84*, T. Beth, N. Cot, and I. Ingemarsson, eds., Springer (New York, 1984), 169–182.

[30] Rivest, R. L.; A. Shamir; and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *CACM*, **21** (February 1978), 120–126.

[31] Roads, Curtis; and John Strawn, eds., *Foundations of Computer Music*, MIT Press (Cambridge, MA, 1985).

[32] Schickele, Professor Peter, *The Definitive Biography of P. D. Q. Bach (1807–1742)?*, Random House (New York, 1976).

[33] Sherman, Alan T., "Little-omega-DTIME and related extensions to DTIME notation," Computer Science Department, University of Maryland Baltimore County (July 1990), in preparation.

[34] Steele, Guy L. Jr., "The Telnet song ('Control-uparrow Q.')," *CACM*, **27**:4 (April 1984), 347–348.

[35] *Webster's Ninth New Collegiate Dictionary*, Merriam-Webster (Springfield, MA, 1983).

[36] Wunderlich, Marvin C., "Computational methods for factoring large integers," *Abacus*, **5**:2 (winter 1988), 19–33.

Records, tapes, compact disks, and video cassettes

[37] "An Evening wasted with Tom Lehrer," Reprise RS–6199 (1959). [LP recording]

[38] Emery, Vince, producer, "The funniest computer science songs," Vince Emery Productions, VE–05 (1989). P.O. Box 460279, San Francisco, CA 94146. [Cassette tape]

[39] "My son, the greatest: The best of Allan Sherman," Rhino Records R2 75771 (1988). [Compact disk recording]

[40] "Songs by Tom Lehrer," Reprise RS–6216 (re-recording, 1966). [LP recording]

[41] "Tom Lehrer: That was the year that was," Reprise RS–6179 (1965). [LP recording]

[42] *Walt Disney's Marry Poppins*, Walt Disney Home Video 023–5 (1980). [VHS video cassette]

[43] "Walt Disney's *Marry Poppins* original motion picture soundtrack" with Julie Andrews and Dick Van Dyke (music and lyrics by Richard M. Sherman and Robert B. Sherman; arranged and conducted by Irwin Kostal), digitally remastered, Walt Disney Records CD–016. [Compact disk recording]

Disclaimer: The character Maria Poopings, all purported circumstances surrounding the alleged discovery of her 1824 manuscript, and all suggested relationships between her and other people are entirely fictional.

A Performance Notes

For anyone foolish enough to attempt singing Poopings's manuscript, Tables 1–2 give an injection from the lyrics into the tune. Table 1 defines the injection for beats 0 through 8; Table 2 defines the injection for the remaining beats. Please note that the melody is slightly different for the verses than for the chorus. For example, whereas the chorus begins on a downbeat, each verse begins on an upbeat. Also, the penultimate musical syllable of each line of the chorus is long (beats 13–14), but that of each line of every verse is short (beat 12).

Table 1: Injection from lyrics into tune for beats 0 through 8.

[15] Alternately, the first line of the chorus can be sung "Sup-er-pol-y-log-a-rith-mic-subex-po-nen-tial-functions!" with the prefix "sub" functioning as a grace note.

Table 2: Injection from lyrics into tune for beats 9 through 15.

			BEATS:			
9	10	11	12	13	14	15
			CHORUS:			
ex-	po-	nen-	tial	func-	-	tions![15]
slower	than	ex-	po-	nen-	-	tial.
tru-	ly	quint-	es-	sen-	-	tial.
ex-	po-	nen-	tial	func-	-	tions!
			VERSE 1:			
Bob	when	Eve's	eaves-	dropping,		
not	a	one-	key	mapping.		
raise	it	to	the	e;		
keep	these	se-	cret-	ly.		Oh!
			VERSE 2:			
appears	to	be	se-	cure:		
one	over	poly-	log	pure.		
least	ex-	po-	nen-	tial,		
splits	n	subex-	po-	nential-	(ly).	Oh!
			VERSE 3:			
lot	of	en-	er-	gy;		
bet-	ter	strat-	e-	gy.		
in-	fo	is	not	doomed,		
po-	wer	be	con-	sumed.		Oh!
			VERSE 4:			
run	a	pro-	gram	P,		
do	so	rever-	si-	bly.		
space	is	ex-	po-	nential,		
real-	ly	is	es-	sential!		Oh!
			VERSE 5:			
can-	not	find	the	way,		
got	a	lot	to	say.		
you	could	fail	the	test.		

BEATS:								
0	1	2	3	4	5	6	7	8
				CHORUS:				
	Sup-	er-	pol-	y-	log'-	rith-	mic	sub-
	Fast-	er	than	a	pol-	y-	log	but
	Ev-	en	though	they're	hard	to	say,	they're
	Sup-	er-	pol-	y-	log'-	rith-	mic	sub-
				VERSE 1:				
For	Alice	to	send	a	mes-	sage	through	to
do	use	a	trap-	door	one-	way	func-	tion—
First	take	a	mes-	sage	x,	let's	say,	and
then	mod	it	out	by	p	times	q	but
				VERSE 2:				
The	pro-	cess	takes	but	pol-	y-	time	and
why	ev-	en	just	a	sin-	gle	bit	is
Though	Alice	thinks	that	Eve	must	spend	time	at
by	us-	ing	Len-	stra's	ellip-	tic	curves,	Eve
				VERSE 3:				
Most	com-	pu-	ta-	tions	dis-	si-	pate	a
we	remove	the	heat	with	water	but	there's	a
Since	thermo-	dy-	namics	does	not	ap-	ply	when
the	laws	of	phys-	ics	don't	re-	quire	that
				VERSE 4:				
Now	Ben-	nett	said	in	seven-	ty-	three,	to
you	sim-	u-	late	the	pro-	gram	P,	but
The	prob-	lem	with	this	meth-	od	is	that
so	trade	off	time	to	save	on	space—	this
				VERSE 5:				
If	you	are	in	an	oral	ex-	am	and
just	sum-	mon	up	these	words	and	then	you've
But	bet-	ter	use	them	care-	ful-	ly	or

B Definitions, Examples, and Notational Digression

In this appendix we give a precise definition of what it means for a function to be superpolylogarithmic subexponential. In addition, we illustrate this definition with concrete examples and explain some of its subtleties. We also define the complexity class SupPolyLog/SubExp, which consists of all languages that can be accepted within deterministic superpolylogarithmic subexponential time; curiously, this class cannot be conveniently described using existing terminology. To define the class SupPolyLog/SubExp in a simple and natural way, we propose and use novel extensions to the popular DTIME notation from complexity theory. *Warning: this section contains little humor and has some educational and scientific value.*

B.1 Definition of Superpolylogarithmic Subexponential

Definitions 1–3 define what it means for a function to be superpolylogarithmic subexponential in terms of the two component notions of superpolylogarithmic and subexponential. For simplicity, we state these definitions for any nonnegative function f that maps the positive integers to the nonnegative real numbers. Informally, Definition 1 says that f is *superpolylogarithmic* if, asymptotically, f grows strictly faster than any polynomial of the logarithm $\log_b n$, for every real base $b > 1$. Similarly, Definition 2 says that f is *subexponential* if, asymptotically, f grows strictly slower than the exponential B^n, for every real base $B > 1$. Finally, Definition 3 says that f is *superpolylogarithmic subexponential* if f is both superpolylogarithmic and subexponential. Additional useful growth rates such as *superlogarithmic*, *superpolynomial*, and *superexponential* can be similarly defined. By introducing absolute values around $f(n)$ in Definitions 1–2, it would be possible to extend these definitions to arbitrary real-valued functions.

Definition 1.
Let f be any function that maps the positive integers to the nonnegative real numbers. The function f is *superpolylogarithmic* if and only if, for every positive integer k, for every real number $b > 1$, and for every real number c, there exists some natural number n_0 such that $f(n) > c(\log_b n)^k$, for all positive integers $n \geq n_0$. $\Box$

To understand Definition 1, it is helpful to be aware of the following two points. First, because the integer k is universally quantified, eliminating the constant c yields an equivalent definition. Second, fixing the base of the logarithm to any particular real number $b > 1$ (for example, fixing $b = 2$) results in an equivalent definition. Fixing the base of the logarithm does not affect Definition 1 because, for every real number $b > 1$, $\log_b = \lg_2 / \lg_2 b$ and $\lg_2 b$ is a constant.

Definition 2.
Let f be any function that maps the positive integers to the nonnegative real numbers. The function f is *subexponential* if and only if, for every real number $B > 1$ and for every positive real number d, there exists some natural number n_0 such that $f(n) < dB^n$, for all positive integers $n \geq n_0$. $\Box$

As with Definition 1, eliminating the constant d in Definition 2 yields an equivalent definition. Unlike Definition 1, however, fixing the base of the exponential in Definition 2 to any real number $B > 1$ (in particular, fixing $B = 2$) semantically alters the definition. Fixing the base of the exponential affects Definition 2 because, for example, 2^n grows strictly slower than 3^n but strictly faster than $(1.5)^n$.

Definition 3.
Let f be any function that maps the positive integers to the nonnegative real numbers. The function f is *superpolylogarithmic subexponential* if and only if f is both superpolylogarithmic and subexponential. $\square$

The standard little-oh and little-omega asymptotic notations[16] provide an equivalent simpler way to characterize superpolylogarithmic subexponential functions. Namely, f is *superpolylogarithmic* if and only if, for every positive integer k, $f(n) \in \omega((\lg n)^k)$. Similarly, f is *subexponential* if and only if, for every real number $B > 1$, $f(n) \in o(B^n)$. Thus, f is *superpolylogarithmic subexponential* if and only if, for every positive integer k and for every real number $B > 1$, $f(n) \in \omega((\lg n)^k) \cap o(B^n)$. That is, f is *superpolylogarithmic subexponential* if and only if, for every positive integer k and for every real number $B > 1$, the limits $\lim_{n\to\infty} (\lg n)^k / f(n)$ and $\lim_{n\to\infty} f(n)/B^n$ exist and equal 0. The equivalence of these simpler definitions with Definitions 1–3, respectively, can be verified using the definitions of limit, little-oh, and little-omega.

Although many authors use the terms "superpolynomial" and "subexponential," few authors define these terms precisely. Unfortunately, this practice obfuscates writing since not all authors have the same definition in mind (if any at all). For example, some authors say that a function f is subexponential if and only if f satisfies Definition 2; others require only that $f(n) \in o(2^n)$.

B.2 Examples

Each of the following three functions is superpolylogarithmic subexponential:

$$f_1(n) = \left(\frac{n}{\lg n}\right)^{0.001}, \qquad f_2(n) = n^2, \qquad \text{and} \qquad f_3(n) = 99^{\sqrt{n \lg n}}, \tag{3}$$

where $f_1(n)$, $f_2(n)$, and $f_3(n)$ are defined for all positive integers n.

Some people might be tempted to conclude that every function is either superpolylogarithmic, subexponential, or both. The following function g, however, is neither superpolylogarithmic nor subexponential:

$$g(n) = \begin{cases} (\lg n)^5 & \text{if } n \text{ is even} \\ (1.01)^n & \text{if } n \text{ is odd,} \end{cases} \tag{4}$$

where $g(n)$ is defined for all positive integers n. Researchers who are disturbed by the "gap" between superpolylogarithmic and subexponential growth rates illustrated by the function g might prefer to adopt a more lenient alternate definition that would require the inequality in Definition 1 to hold only for infinitely many values of n, in the spirit of a lenient alternate definition of big-Omega used by some authors [2, p. 18][9, p. 39].

[16] Let f and g be any functions that map the positive integers to the nonnegative real numbers. By definition [9], $f \in o(g)$ if and only if $\lim_{n\to\infty} f(n)/g(n) = 0$. Similarly, $f \in \omega(g)$ if and only if $\lim_{n\to\infty} f(n)/g(n) = \infty$.

B.3 Notational Digression: The Complexity Class SuperPolyLog/SubExp

Let *SuperPolyLog/SubExp* denote the class of languages that can be accepted within deterministic superpolylogarithmic subexponential time. As will soon be made precise, by the phrase "the language L can be accepted within superpolylogarithmic subexponential time," I mean both that it is possible to accept L in subexponential time and that it is necessary to spend superpolylogarithmic time to accept L. In addition, let *PolyLog* and *SubExp* refer to the class of languages that can be accepted in deterministic polylogarithmic[17] time and deterministic subexponential time, respectively. Although the upper-bound class PolyLog can be easily described using the standard DTIME notation [15] from complexity theory, I know of no convenient existing notation to describe the lower-bound class SuperPolyLog nor the upper-bound class SubExp. In the rest of this section, we explain why DTIME notation is inadequate for defining the class SuperPolyLog/SubExp. We also propose expressive extensions to DTIME notation to overcome these inadequacies and precisely define the class SuperPolyLog/SubExp using our new notation.

Before proceeding, please note that the class SuperPolyLog/SubExp is nonempty. In particular, by the hierarchy theorems for deterministic time [15, pp. 295–296], there exist languages in SuperPolyLog/SubExp $\cap$ P.

The standard DTIME notation from complexity theory is inadequate for describing the class SuperPolyLog/SubExp for the following two reasons. First, DTIME notation is based in essence on big-Oh bounds rather than on little-oh bounds and hence defines what topologists would describe as a "closed" [27] complexity class. SuperPolyLog/SubExp, however, is an "open" complexity class in that it excludes languages on the "boundary" of the class whose complexities are only polylogarithmic or exponential (and not *super*polylogarithmic or *sub*exponential). Second, DTIME describes upper-bound classes rather than lower-bound classes. Some complexity theorists might attempt to circumvent the problem of describing the lower-bound complexity class SuperPolyLog by referring instead to the complement of the related upper-bound class PolyLog. In light of the somewhat sneaky and artificial example g given in equation 4, however, SuperPolyLog does not equal the complement of PolyLog. Thus, the class SuperPolyLog/SubExp is notationally intractable in the sense that it cannot be conveniently defined using existing notation.

To overcome these inadequacies, I propose that the following convenient extensions to DTIME notation be adopted: *O-DTIME*, *o-DTIME*, *Ω-DTIME*, and *ω-DTIME* (pronounced "big-Oh-DTIME," "little-oh-DTIME," "big-Omega-DTIME," and "little-omega-DTIME," respectively). For example, the class ω-DTIME($\lg n$) describes the open lower-bound class of languages L such that: for every Turing Machine M, if M accepts L, then M runs in deterministic time $\omega(\lg n)$. Similarly, the class o-DTIME(2^n) describes the open upper-bound class of languages L such that: for some Turing machine M, M accepts L and M runs in deterministic time $o(2^n)$. The notations *Ω-DTIME* and *O-DTIME* describe corresponding closed classes based on big-Omega and big-Oh bounds, respectively. In light of the linear speed-up theorem for Turing machines [15, pp. 290–291], it follows that O-DTIME($t(n)$) is a pseudonym for DTIME($t(n)$), for all time bounds $t(n)$ such that $\liminf_{n\to\infty} t(n)/n = \infty$. One advantage of these new notations is that they are nondiscriminatory: they treat upper, lower, open, and

[17]A function is *polylogarithmic* if and only if it is of the order $O(\lg n)^k$ for some positive integer k.

closed bounds equally.

Using these new extensions to DTIME notation, the class SuperPolyLog/SubExp can be succinctly defined by

$$\text{SuperPolyLog/SubExp} = \text{SuperPolyLog} \cap \text{SubExp}, \tag{5}$$

where

$$\text{SuperPolyLog} = \bigcap_{k=1}^{\infty} \omega\text{-DTIME}\left((\lg n)^k\right) \tag{6}$$

and

$$\text{SubExp} = \bigcap_{k=1}^{\infty} o\text{-DTIME}\left(\left(1 + \frac{1}{k}\right)^n\right). \tag{7}$$

For a more long-winded explanation of the inadequacies of DTIME notation and of my proposals for overcoming these inadequacies, see my companion paper [33].

About the Author

Alan T. Sherman grew up in Williamsburg, Virginia, where in 1962, he graduated from Mrs. Smith's Kindergarten. Despite playing flute in the Brown University Ice-Skating Band for four years, in June 1978, he received an Sc.B. in mathematics, *magna cum laude*. As part of his graduate studies at the Massachusetts Institute of Technology, he completed a graduate minor in computer music. He also earned a S.M. in electrical engineering and computer science in June 1981, and a Ph.D. in computer science in February 1987. Currently, he is an assistant professor of computer science at the University of Maryland Baltimore County and a member of the University of Maryland Institute for Advanced Computer Studies. In addition to musical computer science, his research interests include algorithmics, cryptology, and VLSI layout algorithms.

Content Indicators[18]

General Terms: Algorithms, Theory.

Categories and Subject Descriptors:

- A. General Literature
  - A.m Miscellaneous
- F. Theory of Computation
  - F.0 General
  - F.1 Computation by Abstract Devices
    - F.1.0 General
    - F.1.3 Complexity Classes
  - F.2 Analysis of Algorithms and Problem Complexity
    - F.2.0 General
    - F.2.2 Nonnumerical Algorithms and Problems

[18] From the classification system used by *Computing Reviews*.