Previous Research

My research interests are at the intersection of theory of computation and mathematics, especially logic. My research across three different projects over the past two years has explored a variety of points in this intersection, and, avoiding premature specialization, spans topics in both sides of the "Theory A" / "Theory B" divide.

Small Circuits and Random Strings In Summer 2011 I got my first taste of research at the DIMACS REU at Rutgers University. For eight weeks there and several months afterwards I worked with Eric Allender and his collaborators to characterize complexity classes defined by reducibility to the set of Kolmogorov-random strings, culminating in [3].

Background: There is a growing body of evidence to suggest a useful connection between the set of Kolmogorov-random strings and the complexity classes P, NP, PSPACE, NEXP, $\mathsf{EXPSPACE}$, and BPP. We need some definitions; see [4] for details. Where U is a universal Turing machine, let $C_U(x)$ be the length of the shortest d so that $C_U(d) = x$; thus C_U provides a measure of compressibility. A somewhat better-behaved measure K_U is analogous but for a technical constraint on the machine U. A string is Kolmogorov-random if $K_U(x) \geq |x|$. We denote this set R_{K_U}.

It is known that R_{K_U} is Turing-equivalent to the halting problem, which makes it at first very surprising that relativizations of common complexity classes to R_{K_U} could look anything like complexity classes. However, known *efficient* (i.e., polynomial) reductions from the halting problem to R_{K_U} are nonuniform (i.e., are computed with circuits rather than machines). This is the first hint that something like $\mathsf{P}^{R_{K_U}}$ might be of interest to a complexity theorist. It turns out that further technical restrictions must be made in order to get something that looks like a complexity class (in particular, we must do something to get rid of the undecidable sets that may remain in $\mathsf{P}^{R_{K_U}}$). Varying these technical restrictions results, surprisingly, in a variety of classes which can be positioned among the common complexity classes named above. See [2] for details.

Our Work: We concerned ourselves with one such class, $\mathscr{C}$, for which it is known that $\mathsf{BPP} \subseteq \mathscr{C} \subseteq \mathsf{PSPACE}$. We conjecture that the upper bound on $\mathscr{C}$ can be improved to $\mathsf{PSPACE} \cap \mathsf{P}/\mathsf{poly}$.

During the Summer I worked with Professor Allender to extend existing incompressibility-based techniques and improve partial results towards the conjectured upper bound. I discovered several novel extensions to the existing techniques, which resulted in a number of technical structural results characterizing the nature of advice strings that might be used in a P/poly upper bound. The work became more collaborative in the Fall when a surprising connection to the proof theory of certain extensions to Peano Arithmetic surfaced and, concurrently, two new collaborators joined the project. The shift taught me about the occasional necessity of making large changes to a research approach and about how to collaborate across locations and time zones.

In the end, we used the connection to proof theory to establish a stronger conjecture, conditioned on the provability of certain true sentences in particular extensions of Peano Arithmetic. Later work by Buhrman and Loff, while supporting our conjectured upper bound, disproves our stronger conjecture, thus transfering our result to an independence theorem for Peano Arithmetic.

Broader Impacts: If, as seems likely, $\mathscr{C}$ can be upper-bounded by $\mathsf{PSPACE} \cap \mathsf{P}/\mathsf{poly}$, then the lack of complexity classes between BPP and $\mathsf{PSPACE} \cap \mathsf{P}/\mathsf{poly}$ motivates the conjecture $\mathscr{C} = \mathsf{BPP}$, a proof of which could provide a deep connection between derandomization and Kolmogorov complexity. This in turn raises the question: could we use this characterization of BPP towards a proof of $\mathsf{P} = \mathsf{BPP}$? Some of the bounds in the area (though not the containment $\mathsf{BPP} \subseteq \mathscr{C}$) do use nonrelativizing techniques, although it seems that new nonrelativizing techniques would be needed to achieve $\mathsf{P} = \mathsf{BPP}$. Progress here, however, would potentially vastly deepen our understanding of computation in general. Again, see [2] for details.

Language Support for Declarative Web UIs In Summer 2012 I interned on Google's Dart team, led by Lars Bak, creator of Java's Hotspot and Chrome's V8, state-of-the-art just-in-time compilers.

Background: Dart is a new client-and-server-side language for the web, designed to address many of the problems encountered when writing large web applications in Javascript (JS). These include slow startup, since the object hierarchy must be initialized imperatively, poor maintainability and support for code management tools , since JS has no type annotations or static type checking, and poor code reusability, since the lack of a built-in class-based object system has led to many conflicting implementations of class systems. Dart addresses these problems and more, while working within the constraint that it must run both on its own virtual machine and compile well to JS. In addition, Dart is the first mainstream language with optional static typing, bringing the advantage that types can be written where they are useful for code readability or performance and need not be written elsewhere.

My Work: The lack of encapulation in HTML, together with the many conflicting implementations of class systems in JS, has led to fragmentation and poor encapsulation in the world of JS UI frameworks. This has led browser teams to propose modifications to the underlying platform. I worked to adapt the very early prototype JS platform modifications allowing well-encapsulated HTML-based widgets in [1] to something which fit Dart's class-based object orientation. This required designing new code generation in the `dart2js` compiler to allow Dart classes to extend browser-native objects. In the process, I contributed to the design of Dart's main client-side web programming library, designed the isolation and encapsulation model for Dart UI components, and contributed to a templating language for Dart data binding. All of these now have significant user bases. The work was highly collaborative, with frequent design meetings guiding development.

Broader Impacts: Dart has the potential for enormously broad impact by drastically lowering the engineering cost to build large, web-based applications with rich and responsive UIs. Indeed, achieving responsive UIs for web-based apps has been a stumbling block to their wider development, and has traditionally required convoluted programming techniques available only to large companies—Google, Microsoft, etc.—able to devote vast engineering resources to the design and upkeep of their products. By lowering that barrier, Dart in general and Dart's UI framework in particular will help to bring web-app development to much more of the engineering world, enabling applications in areas with historically lower profit margins but great societal impact: healthcare systems, nonprofit support, etc. These applications will bring with them all the attendant benefits of the web, e.g., mobile accessibility and data security.

Lower Bounds in Communication Complexity In Fall 2011 I began work with Professor Paul Beame at the University of Washington. After an eight-month reading course on a variety of topics in complexity, in Spring 2012 we began working on lower bounds in communication complexity. For details on this work and ideas stemming from it, see my research proposal.

References

[1] Introduction to web components. `http://dvcs.w3.org/hg/webcomponents/raw-file/tip/explainer/index.html`. Accessed: 10/28/2012.

[2] E. Allender. Curiouser and curiouser: The link between incompressibility and complexity. *Proc. Computability in Europe*, 2012.

[3] E. Allender, G. Davie, L. Friedman, S. B. Hopkins, and I. Tzameret. Kolmogorov complexity, circuits, and the strength of formal theories of arithmetic. *Submitted.*

[4] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, 3rd edition, 2008.