Personal Comments about Peter Andrews

Dale Miller

17 June 2025

I want to add more personal observations about Peter and his work than what appears in the *In Memoriam*.

1 Being a Ph.D. Student with Peter Andrews

An important research project for Peter was constructing a theorem-proving environment for the version of higher-order logic that his Ph.D. advisor, Alonzo Church, designed and named the Simple Theory of Types. Peter began working on this theorem-proving project around 1974 and utilized the then-popular AI programming language LISP. When I applied to the Mathematics Department at Carnegie Mellon University in 1978, I mentioned in my application that I had had a summer job programming in LISP. As a result, I was a natural pick for Peter to fill a research assistant position on his NSF-funded theorem-proving project. While Peter signed a couple of Ph.D. theses before I arrived at CMU, I was the first student to graduate after working with him on this theorem-proving project.

Anyone who worked with Peter quickly recognized his kindness, patience, and meticulous attention to detail. Two historically significant instances highlight this diligence with details:

- As a Ph.D. student in the early 1960s, Peter was already interested in automating proof. He identified a serious gap in the proof of a theorem from Jacques Herbrand's 1929 Ph.D. thesis, then one of the few notable results in computational logic. After corresponding with Harvard University Philosophy professor Burton Dreben, an expert on Herbrand's work, and a car trip from Princeton to Harvard, Peter convinced Dreben of the error. Together with Stål Aanderaa, they published [10] an explicit counterexample to a lemma in Herbrand's thesis. Dreben later corrected Herbrand's proof. For more on this correction, see Andrews [3] and Dreben [9].

- Peter discovered that Leon Henkin's definition of general models for higher-order logic [11] allowed for non-extensional models [1], which is problematic since the theory was intended to be extensional. Fortunately, the solution was relatively straightforward. Henkin's flaw was in fixing the semantics of propositional and quantification connectives, an insufficient foundation for true higher-order logic. Andrews demonstrated that by fixing the semantics of equality at all finite types, the semantics of propositional constants and quantifiers are also fixed, ensuring only extensional models are possible.

As Peter's student, this bit of history meant two things to me. First, the topic of mathematical logic was full of details that even bright people can get wrong. Second, I had to be very careful when writing my dissertation before giving it to Peter for a critical reading. Ultimately, I kept the technical core of my dissertation as concise as possible and devoted as much time as I could to the technical details. I am pleased that Peter never reported any errors in my dissertation.

I can share another couple of anecdotes. Peter kept a journal composed of loose pages. He recommended that I do the same, and he strongly suggested that I follow his habit of putting the current millennium, century, decade, year, month, day, hour, and minute at the top of every page. I do date my loose pages, but in an act of defiance, I use a more common style for dates. Once, when we were writing a conference paper together, I asked him if we could stop writing "well-formed formula" (and the abbreviation wff) and write just "formula" since ill-formed formulas were of no interest in that paper. He insisted on keeping the wordy alternative, finally justifying that choice with the claim that "if that terminology was good enough for Church, it is good enough for me."

2 Introducing Peter Andrews's Work to Proof Theorists

Peter research did not overlap much with the topic of structural proof theory. Although he used natural deduction-style proofs (à la Fitch) for interacting with his theorem prover, the automation of that prover relied on techniques such as unification, skolemization, resolution, and negation normal forms [5]. However, I can think of two reasons that the proof theory community should care about some of his research results and ideas.

2.1 First-Order and Higher-Order Quantification

People working in structural proof theory often dismiss first-order quantification as not being particularly interesting. Sometimes, they embrace the expressiveness and challenges of second-order quantification. However, a more general setting for quantification was established by Church in his paper on the Simple Theory of Types [8]. In that setting, quantification at all (finite) simple types was possible, and term structures were identified with simply typed λ-terms. While early papers by Henkin, Schütte, Takahashi, Girard, Andrews, and others established the most basic proof-theoretic properties of Church's logic, this higher-order logic is not often examined by the proof theory community today.

I learned Church's approach to quantificational logic from Andrews and from implementing aspects of it in his theorem prover. I gradually came to understand that a direct mixing of Church's approach to encoding terms and formulas as simply typed λ-terms with Gentzen's sequent calculus yields a powerful and elegant proof system for quantificational logic. In particular, the sequent calculus supports a natural notion of *binder mobility* where λ-abstractions within terms can move to quantified bindings in formulas, which, in turn, can move to eigenvariable bindings in proofs. The first programming language to explicitly represent this mobility of binding was λProlog [14], and it is the root of its declarative treatment of bindings. The Abella proof assistant [6] also exploits and extends this same kind of binder mobility.

I strongly recommend Peter's textbook [4] to anyone interested in learning the basics of first-order and higher-order quantificational logic. My first class in logic was taught by Peter in 1978 from a mimeograph copy of an early edition of that book. Later, I taught logic using the first edition of the textbook. The second edition of this book is still available for purchase. Andrews's Festschrift [7] includes historical and more modern contributions surrounding higher-order logic.

2.2 Matings as the Essence of Proof

For Peter, logic meant classical logic, and his favorite proof structure was something he called *matings* [2]. This structure consisted of pairs of complementary occurrences of literals within propositional formulas and provided a compact certificate of provability. For Peter, matings were more than merely a certificate for proof: they were the *essence* of proof. For a short time, I was seduced by the promise of matings, as they offered a more parallel proof structure than the sequential structures that can be tedious when building sequent calculus proofs or resolution refutations. I attempted to employ them in my dissertation [13] to define a "focused construction" of natural deduction proofs (no relation to the modern notion of a focused proof system). However, I soon gave up on using matings since all the operations I attempted to do with them had exponential costs, which amounted to the same cost as if there were no explicit mention of matings. It is exciting to see the work on combinatorial proofs [12, 15], since I can imagine that this recent work might provide some validation for Peter's convictions about matings.

References

[1] Peter B. Andrews. General models and extensionality. *Journal of Symbolic Logic*, 37:395–397, 1972.

[2] Peter B. Andrews. Theorem proving via general matings. *J. ACM*, 28(2):193–214, 1981.

[3] Peter B. Andrews. Herbrand award acceptance speech. *J. of Automated Reasoning*, 31:169–187, 2003.

[4] Peter B. Andrews. *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof.* Springer, 2014.

[5] Peter B. Andrews, Matthew Bishop, Sunil Issar, Dan Nesmith, Frank Pfenning, and Hongwei Xi. TPS: A theorem proving system for classical type theory. *Journal of Automated Reasoning*, 16(3):321–353, 1996.

[6] David Baelde, Kaustuv Chaudhuri, Andrew Gacek, Dale Miller, Gopalan Nadathur, Alwen Tiu, and Yuting Wang. Abella: A system for reasoning about relational specifications. *J. of Formalized Reasoning*, 7(2):1–89, 2014.

[7] Christoph Benzmüller, Chad E. Brown, Jörg Siekmann, and Richard Statman, editors. *Reasoning in simple type theory — Festschrift in honor of Peter B. Andrews on his 70th birthday*, volume 17 of *Studies in Logic*. College Publications, 2008.

[8] Alonzo Church. A formulation of the Simple Theory of Types. *J. of Symbolic Logic*, 5:56–68, 1940.

[9] Burton Dreben. Herbrand's thesis. In Jean van Heijenoort, editor, *From Frege to Gödel: A Source Book in Mathematical Logic, 1879–1931*, pages 525–529. Harvard University Press, 1967.

[10] Burton Dreben, Peter Andrews, and Stål Aanderaa. False lemmas in Herbrand. *Bulletin of the American Mathematical Society*, 69:699–706, 1963.

[11] Leon Henkin. Completeness in the theory of types. *Journal of Symbolic Logic*, 15(2):81–91, 1950.

[12] Dominic J. D. Hughes. Proofs without syntax. *Annals of Mathematics*, 143(3):1065–1076, November 2006.

[13] Dale Miller. *Proofs in Higher-order Logic.* PhD thesis, Carnegie-Mellon University, August 1983. University of Pennsylvania technical report MS-CIS-83-37.

[14] Dale Miller and Gopalan Nadathur. *Programming with Higher-Order Logic.* Cambridge University Press, June 2012.

[15] Lutz Straßburger. Combinatorial flows and their normalisation. In *2nd International Conference on Formal Structures for Computation and Deduction (FSCD 2017).* Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2017.