In memory of Seymour Ginsburg 1928 – 2004

Seymour Ginsburg, a pioneer of formal language theory and database theory, passed away on December 5 after a long battle with Alzheimer's disease.

Seymour received his B.S. from City College of New York in 1948 and his Ph.D. in Mathematics in 1952 from the University of Michigan. From 1951 to 1955, he was Assistant Professor of Mathematics at the University of Miami (Florida). He started working in Computer Science in 1955 when he moved to the Northrop Corporation, and later at NCR, Hughes Aircraft, and System Development Corporation. In 1966, he joined the University of Southern California as a professor and helped establish the Computer Science Program in 1968.

The author of over 100 research papers and three books, Seymour Ginsburg was a leader of theoretical computer science. In the late 50s and 60s, Seymour was one of the giants of automata and formal language theory. While Noam Chomsky introduced context-free grammars to model natural languages, Seymour was the first to note the connection between context-free languages and “Algol-like” computer languages specified in Backus normal form, thus bringing formal language theory at the center of programming language research. His results on context-free grammars and push-down acceptors are some of the deepest and most beautiful in the area, and remain standard tools for many computer scientists.

One of the constant themes in Seymour's research in formal language theory was the unification of three different views of languages: grammar-based, acceptor-based, and algebraic. This culminated in his work with Sheila Greibach on Abstract Families of Languages (AFLs), in which they were joined by a long list of researchers including Goldstine, Harrison, Hopcroft, Spanier, and many others. In a nutshell, AFL theory characterizes the families of languages that can be defined by “reasonable” accepting devices purely in terms of their closure properties (e.g. closure under intersection with regular languages, concatenation, union, inverse homomorphism, Kleene +, etc). For example, one would expect any family of languages defined by acceptors whose control involves finite-state transitions to be closed under intersection with regular languages, since any finite-state control can be naturally intersected with an FSA. More surprising and elegant is the converse: if a family of languages has certain closure properties, then it must also have a “reasonable” corresponding accepting device.

In the 70s, Seymour developed the theory of grammar forms together with his PhD student Armin Cremers. These provide a mechanism for defining families of grammars whose rules have particular shapes, similarly to normal forms. Grammar forms are more flexible than usual grammars, and better behaved. For example, equivalence of grammar forms is decidable. This result is based on an elegant prime decomposition theorem for families of languages definable by grammar forms, similar to the prime decomposition theorem for the integers.

In the 80s, Seymour was one several senior theoreticians to make a transition into an emerging area that they viewed as an exciting new frontier: database theory. Seymour embarked upon this task with determination and enthusiasm, and never looked back to the familiar shores of language theory. Together with students and collaborators, he soon started producing what was to be a host of insightful results on functional dependencies, object histories and historical databases, datalog, and order dependencies. Already in 1981, he and his newly minted database group had a visible presence at the XP2 workshop, the early informal meeting of database theoreticians, with four papers. In 1982, he organized the first PODS conference in Marina del Rey. Seymour continued to be active in database theory and to attend PODS into the 90s. His last paper, on data restructuring, was published in 1999.

For his 64th birthday, Seymour was honored with a special surprise session at PODS, and a festschrift in his honor was edited by Jeff Ullman. Seymour Ginsburg's passing away is a loss to Computer Science and to the many whose lives he has touched.

S. Abiteboul, R. Hull, V. Vianu

Personal reminiscences

Sheila Greibach, University of California, Los Angeles

I was deeply saddened to hear of Seymour's death, and yet a smile came as I remembered him, as he would probably have preferred to be remembered, when I first met him. He came bounding into my office in the basement of the then Aiken Computer Lab at Harvard, vigorous, energetic, self-confident, to tell me how he just had to meet the “girl who wrote that great thesis” he was told about. I was about 24 and thought of myself as a “girl” and was altogether flattered to have a “senior” (thirtysomething) researcher aware of my thesis.

There followed a very productive collaboration, during which I consulted at SDC in Santa Monica summers and also part of a sabbatical. We drank truly awful machine coffee, discussed our papers, and Seymour held forth as he loved to do on the research game –

how to get topics, how to get papers published, and how to get known. One time I thought I was getting caffeine jitters, but it turned out to be a mini-quake, my first such.

Seymour enjoyed his role as mentor to younger researchers. As a consequence, I met many young computer scientists through him. I collaborated (separately) with Seymour and three of them – Michael Harrison, John Hopcroft, and Jonathan Goldstine – despite Seymour's strictures against 3 authors on a paper – he felt that 3 people working in pairs meant 3 papers but as a trio, only 1. I remember going in Mike's little sports car to Seymour's Van Nuys home one summer afternoon – we brought him a bottle of champagne but he preferred KoolAid. We swam in his pool and heard him explain about “swimming pool” theorems – results major enough that the resulting promotion or raise enabled one to purchase a house with a swimming pool.

Seymour had firm ideas on many things. For one, meat was for dinner, not lunch, so working lunches meant an Italian restaurant – an exception being made for the marvelous Chinese lunches Gene Rose sometimes arranged (where I learned NOT to use chopsticks since in that crowd it required my greater speed with a fork in order to get a taste of everything).

My collaboration with Seymour ended, oddly enough, about when I moved to Los Angeles to take a tenured position at UCLA and, within the year, marry and obtain my own house-with-swimming- pool. At that time, our research interests started to diverge. We remained in touch off and on. I think the last time I saw him was at the dinner in honor of his 64th birthday, when we contributed to a book in his honor edited by Jeff Ullman. He was still as I like to remember him – smart, funny, overflowing with ideas, vigorous and full of joie de vivre.

Michael A. Harrison, University of California, Berkeley

I was aware of Seymour Ginsburg and his work in automata theory in 1963 when I joined the Berkeley faculty. There were quite a few people at Cal interested in theoretical computer sciences including Dana Scott, Eli Shamir, Ed Spanier, John Rhodes, and Bob Solovay. One of the first things we did was to start a joint seminar. Seymour Ginsburg was one of the first non-local speakers. Seymour and I had a lot of research interests in common. Moreover our styles were compatible.

Seymour was employed at the System Development Corporation in those days. They were a systems company and Seymour was well respected there. He wanted to have a larger research program so he raised funds from the government agencies. He invited colleagues to spend time with him. For academics, this meant the summers or semester breaks. His circle included Sheila Greibach, John Hopcroft, Gene Rose, Ed. Spanier, Joe Ullian, and Jeff Ullman. These were very pleasant and productive collaborations.

Seymour was a fine collaborator and he was very fine mentor to his younger colleagues.

By the late 60’s our research interests were diverging. But I still stayed in touch and we would arrange to visit one another for seminars or other occasions.

Seymour was a devoted family man. I can remember many enjoyable lunches in which we talked about out families.

I am particularly pleased that Susan and I could attend a surprise retirement dinner for Seymour. Sadly, I think that was the last time I saw him. I miss him.

Ellis Horowitz, University of Southern California
Summary of remarks delivered at Seymour Ginsburg's funeral, Dec. 7, 2004

I first met Seymour when I joined the USC faculty in 1973. At that time there was just a small computer science group, and Seymour was the only senior professor. What stands out in my memory was how dedicated he was to mentoring the junior faculty and the Ph.D. students. Seymour would always give freely of his time to talk about computer science, research, publication, grantsmanship, and any topic one wished to discuss. Every new Ph.D. student was ushered into his office, and Seymour would go over the graduation requirements from A-Z.

Of course Seymour was an accomplished researcher and well-known for his work on formal languages and automata theory. I remember the many faculty who would come to USC and visit with him, for a day, a week, or a semester. The year he won his Gugenheim he traveled around the world giving lectures and working with other researchers.

Seymour had many ideas about academic excellence, and though he was not an administrator he took an active role in departmental affairs. He always argued to keep our standards high. But he was always charitable when a student needed more time to finish his Ph.D. research.

My fondest memories of Seymour go back to the years when his son David attended USC. Seymour was a devoted father, and David would often drop by his office to discuss his classes. My office was right next door, so I would drop by occasionally and hear about the great (and the not-so-great) USC faculty teaching elsewhere in the college. When David went on to Medical School in Pittsburgh, Seymour found time to recruit students for USC from CMU and Pitt. The devotion between father and son was wonderful to see, and a side of Seymour that many were unaware of.

Daniel Rosenkrantz, University at Albany

I first met Seymour Ginsburg in the late 1960s, at theory conferences. He was instrumental in moving the switching theory community, with its electrical engineering heritage, in a more abstract direction. For instance, he generalized finite automata by providing a perspective of devices as language processors. Indeed, the switching theory community, as exemplified by the name of its primary conference, changed its name to Switching and Automata Theory (with a subsequent name change to the more general Foundations of Computer Science). In the early 1980s, Seymour was a force in crystalizing a theory-oriented database community, focused on the PODS Conference.

Seymour helped develop a tradition of computer science conferences that were very selective and at a high intellectual level, held in an environment conducive to scientific exchange. He particularly loved the atmosphere at Marina del Rey, and used every opportunity as a local arrangements chair of conferences to hold them there.

Seymour had clarity of vision and a steady compass for intellectual integrity and rigor. He pursued generality, abstraction, and the discovery of inherent structure, and influenced many other computer scientists to do so. He will be sorely missed.

Jeffrey Ullman, Stanford University

The following is a substantial reworking of the preface written for the volume that commemorated Seymour Ginsburg's 2^6-th birthday.

I step off the plane, and there are palm trees all over the place, and grocery stores sell booze, and I feel like I'm on another planet. As a 23-year-old graduate student growing up in New York, who had never been any further from home than Boston, my summer job working for Seymour Ginsburg was a heady experience even before I reported to work. The world was a much bigger place in 1965 than it is today. Jet planes had just come into use, and the cost of a long-distance call, in today's terms, was about $8 per minute. Yet the excitement of living in a new place was exceeded by that of the new intellectual environment in which I found myself: Seymour and his colleagues engaged in bringing rigor to the infant discipline of computer science.

In the summer of 1965, Seymour Ginsburg was running a research project developing language theory in a wing of the System Development Corp., in Santa Monica. Seymour had gathered about him a group of researchers, Sheila Greibach, Mike Harrison, Gene Rose, Ed Spanier, and Joe Ullian, who formed one of a small number of groups developing the foundations of computer science. For that summer, the Ginsburg group also included me, an Electrical Engineering student from Princeton.

Several years before, Seymour had demonstrated the incorrectness of the engineer's intuition about how to minimize automata with don't-care's, and he had developed a rigorous theory for the problem. On the strength of this and other developments, theory was beginning to emerge for other areas of the field, and Seymour was at the center of the group that eventually became the theoretical CS community. The big deal in 1965 was context-free languages. Seymour had created an outline of the important issues to consider when examining a formalism:

- What can and what cannot be described by the formalism?
- What are the algebraic properties (“closure properties”) of the things described by the formalism?
- What can and what cannot be decided about the things described by the formalism (decision properties)?

These three issues still stand today as a guide to new theories as they occur in the many branches of computer science.

I remember having a good deal of difficulty learning the style of computer science theory that summer; my previous training had not prepared me well. But learning to think rigorously and to analyze questions patiently are not the most important things I took away from that summer. The lesson I received from Seymour Ginsburg was a bit more subtle. There is a model of research that is prevalent in engineering fields: read a few papers, look for a way to make a small improvement, write a paper about it, and leave some open questions so someone else will come along, make another small improvement, and reference your paper. I had assimilated this approach not only from my studies, but at a more conventional research lab where I had worked the previous summer.

The Ginsburg group of 1965 had a completely different view of the world. They were out to change how people thought about computing. They were investigators for the sheer joy of discovering something new or resolving a question that seemed hard. They saw research not as a way to earn a living, but as something that should and could make things better in all sorts of ways. They valued the new and the surprising, rather than the incremental.

As I think back over my own career, I wonder how well I transmitted the lessons of 1965 to my own students. Certainly, these students seem one way or another to have gotten these messages pretty well. I wonder if they know the influence of Seymour Ginsburg on their own outlook. Probably not; I never told them, and I should have.

Moshe Vardi, Rice University

I met Seymour Ginsburg in January 1982, when I came to USC to give a colloquium. Seymour was the most senior researcher in database theory. I was rather intimidated by him. It was not just his stern demeanor. Seymour had written books 20 years before I got my PhD. To me, at the time, that put him in the pantheon of computer science gods. It was humbling to give a talk when god is in the audience. Could the theorems proved by us mortals be compared to the theorems proved by the gods themselves?

I quickly discovered that Seymour's gruff demeanor was only on the exterior. Seymour was hospitable and gracious. He took genuine interest in the work of young researchers and encouraged them to pursue their interests with vigor. Like the proverbial Israeli, Seymour was thorny on the outside and sweet on the inside. Perhaps because of that, we developed a warm relationship that lasted for many years.

During the 1980s and 1990s, Seymour's research focus was on databases. While he liked the ``practical'' motivation of database research, he was at heart a mathematician and judged results by their inherent mathematical beauty. In that sense he was a purist, and I appreciated his purism even though I did not always agree with him. When Seymour stopped attending the annual PODS meeting, his absence was felt. His passing always leaves our pantheon with a permanent gap.

Serge Abiteboul, INRIA-Futurs

For me, Seymour was at first the professor with a strange hat and immutable clothes going down the corridor of the department, a bit abrupt but always so helpful to the young ones. Seymour Ginsburg, the famous professor, a pioneer of formal languages, a knight of computer science theory, always found seemingly unlimited time for his PhD students. I remember my visits to his place on Saturdays to go over my thesis together, after the invariable spaghetti lunch. He would check every single definition, result, every single proof, sentence, every single comma. He would talk about academia, very concretely: referee reports, grading complaints, tenure, etc. But he would also talk about esthetics, ethics, psychology, the philosophy of research. Seymour did not take lightly his role as an advisor.

Esthetics... his proof of the pumping lemma in class sticks to mind for some reason. This was truly beautiful, as crystal clear as his books.

After many years, when I look back, I see that no one influenced my work more than Seymour. I realize when I talk to my PhD students, that I am repeating his lessons. When I write a paper, I try to follow his very precise guidelines. When I get started on some new problem, I wonder: what questions would have Seymour asked?

I want to thank him once again for everything he taught me.

Richard Hull, Bell Labs Research, Lucent

“A new model means new questions!” – that was at the heart of Seymour's research, and his most impactful lesson for me. Seymour created abstract models of emerging Computer Science technologies not for the purpose of modeling, but rather to ask, in a precise manner, a question that would cut through the details, and reveal new fundamental properties of the technology.

Another of his axioms was “One must change directions every five years” – an indication of his courage and self-confidence. When I first met Seymour as his post-doc at USC in 1979, he was just abandoning his long history with formal language theory and embarking on the study of a great new model – relational databases and dependencies. And he entered this area with great energy, supervising a post-doc and graduating 4 Ph.D. students in as many years. There was no “resting on his laurels” for Seymour.

Seymour has been a warm and loyal mentor and friend for me, and for many in the family of students and researchers that he worked with. As I continued on at USC as a young professor, Seymour and I co-authored several papers. But after a few years he “cut the chains” and advised me to find my own way. He knew this was crucial for my research success in the long run, even if it meant losing a close collaborator.

Seymour's attention to detail is legendary, a strong counterpoint to the sweeping insights and connections that he found through the years. He spent long hours with me and with his Ph.D. students going over every proof in a co-authored paper or thesis. He insisted on thorough responses to any referee suggestions.

Seymour put the same energy and attention into the department's Ph.D. program. One of the fixtures in the program were the “screening” meetings, in which faculty would evaluate all of the students who completed a year of study, to determine if they were really Ph.D. material. Seymour felt that each student had to be considered in depth – just looking at grades or scores was not sufficient. These marathon sessions must have been agonizing for the students whose positions hung in the balance, and as Seymour lead us through comprehensive discussions for each student, they weren't much better for the rest of us on the faculty.

Seymour combined a deep concern for intellectual rigor with a rich, enduring fatherly love for the growth of individual students and colleagues. It is this combination which I admire most about Seymour – I will always strive to live up to his standard of bringing the highest of human values into one's professional life.

Victor Vianu, University of California, San Diego

When my mother heard about Seymour Ginsburg's death, she searched into a memorabilia-filled shoe box and handed me a letter from him written 25 years ago, when I had just arrived as a graduate student at USC. He was reassuring her that her son, now many thousands of miles away from her, was doing just fine – there was no need to worry. Needless to say, he never mentioned the letter to me. This is the way Seymour was – doing what he thought was right, sometimes behind the scenes if he deemed it appropriate.

Of the many things he taught me as an advisor, perhaps most important was the idealism and sheer joy of research. To him, good research was necessarily beautiful research, and doing it was more than personally satisfying – it was an almost sacred duty, his way of improving the world. Seymour had an uncanny ability to extract the relevant mathematical structure from messy real-world artifacts, and taught me the importance of devising elegant models and asking the right questions. He spent an inordinate amount of time with his students and made sure they learned a myriad of things that would serve them well throughout their careers: the logistics and ethics of research, that quality is better than quantity (his worst possible recommendation: "this person is known for his many papers"), that one should avoid spending time on routine questions ("filling in the matrix", as he put it), that being scooped occasionally is a fact of life that should be accepted graciously.

Seymour also taught me that even giants have their vulnerabilities and insecurities. He used to joke about his nightmare that one day "someone from the deep end of Siberia" would be poring over his old papers and inform him that he had found a bug in some crucial technical lemma. He was uncomfortable in front of a large audience, but shone as a charismatic and contagiously enthusiastic speaker in front of his graduate students, who listened to him in rapture as he unfolded for them the gems of language theory and the high drama of research.

Having been Seymour Ginsburg's student was an enormous privilege, and having known him as a person an endearing and life-shaping experience.