needed to link qualitative reasoning, often used to describe the desirable properties of a system, to the quantitative aspects that emerge when a system or its components are observed with the help of numerically based sensing systems. *Fuzzy logic* provides an answer to some of these issues.

Lotfi Zadeh is widely known as the inventor and "father" of fuzzy logic. The concept of fuzzy logic has received attention in practically all fields of science, engineering, as well as in linguistics, psychology, economics, and in many other fields within the social sciences and humanities. Although the idea formalizes the natural concept of approximate reasoning, it was met with fierce resistance in some quarters for many years. Fuzzy logic has now become a standard tool in the repertoire of methods that are used in engineering fields such as operations research, systems engineering, and problem solving in artificial intelligence. Recent theoretical results, which show that fuzzy logic can be used to approximate continuous functions on a compact set, have served to reconstruct the links between mathematical systems theory, fuzzy logic, **neural networks**, and other areas of computational intelligence.

Zadeh was and continues to be a vocal proponent of the integration of electrical engineering and computer science. A new direction in his work relates to what he calls the computational theory of perceptions. Zadeh is also an advanced amateur photographer specializing in portraiture of prominent personalities. His portraits include those of Alexander Kerensky (1881–1970; the premier of Russia just before the October Revolution), Richard Nixon (1913–94), Harry Truman (1884–1972), **Claude Shannon** (1916–2001), Aldous Huxley (1894–1963), and many others. His multilingual and multicultural background, his long experience in the world of science and engineering, and his life in major international centers such as Berkeley and New York have shaped his very broad humanistic perspective on the role of science and engineering in society.

BIOGRAPHY

Lotfi Asker Zadeh. Born 4 February 1921, in Baku, Azerbaijan. B.S. in electrical engineering from the University of Teheran, 1942; M.S. in electrical engineering from Massachusetts Institute of Technology, 1946; Ph.D. in electrical engineering from Columbia University, 1949. Professor, Columbia University, 1949–59. Professor of electrical engineering at University of California–Berkeley, 1959–present; founding chair of Berkeley's Department of Electrical Engineering and Computer Science, 1967. Visiting professor at MIT, 1963, 1968; visiting member of the Institute for Advanced Studies at Princeton, 1956. Currently, professor in the Graduate School at Berkeley and director of the Berkeley Initiative in Soft Computing. Recipient of numerous honors and awards, including Congress Award from the International Congress on Applied Systems, Research and Cybernetics, 1980; Outstanding Paper Award from the International Symposium on Multiple-valued Logic, 1984; Honda Prize, 1989; Berkeley Citation, from the University of California at Berkeley, 1991; IEEE Richard W. Hamming medal, 1992; and Grigore Moisil Prize for Fundamental Research, 1993.

SELECTED WRITINGS

Zadeh, Lotfi A. "Fuzzy Sets." *Information and Control,* Vol. 8, 1965, pp. 338–353.

———. "A Computational Approach to Fuzzy Quantifiers in Natural Languages." *Computers and Mathematics,* Vol. 9, No. 1, 1983, pp. 149–184.

———. "From Computing with Numbers to Computing with Words—From Manipulation of Measurements to Manipulation of Perceptions." *IEEE Transactions on Circuits and Systems,* Vol. 45, 1999, pp. 105–119.

Zadeh, Lotfi A., and Charles A. Desoer. *Linear System Theory: The State Space Approach.* New York, McGraw-Hill, 1963.

FURTHER READING

Zadeh, Fay. *My Life and Travels with the Father of Fuzzy Logic.* Albuquerque, N.M.: T S I Press, 1998.

—*Erol Gelenbe*

Z1, Z2, Z3, and Z4

From 1936 to 1945 in Berlin, the German inventor **Konrad Zuse** (1910–95) built some of the earliest computers: the machines Z1 (1936–38), Z2 (1940), Z3 (1938–41), and Z4 (1941–45). The Z1 was a mechanical device, whereas the Z3, which had the same logical structure, was built using electrical telephone relays. The Z2 was an intermediate prototype in which the memory consisted of mechanical components and the processor was made of relays. The Z4 was also a relay machine—a "better Z3" in the sense that it had a much greater memory and the instruction set was enlarged to handle more operations.

All machines used the **binary system** internally for the arithmetic operations. Numbers were entered in decimal notation and the processor took care of transforming from the decimal to the binary system. After computing the result using binary arithmetic, it was transformed from binary to decimal and was displayed using lamps.

The Z1, Z3, and Z4 worked with **floating-point** numbers, numbers written as an exponent and a mantissa, as in 1.5×2^3. Zuse wanted his machines to be used for engineering and business computations and developed an internal numerical representation that strongly resembles the floating-point format of today's **Institute of Electrical and Electronics Engineers** (IEEE). Each number was stored using three fields: the sign of the number, the exponent of the number in two's-complement notation, and the mantissa of the number. The number 2.0, for example, was stored as a positive sign, binary exponent 1 and mantissa 1.0 (i.e., $2 = 1.0 \times 2^1$). The processor of the Z1, Z3, and Z4 thus consisted of two main components, one for handing the exponent of a number and one for handling the mantissa.

The three main machines (the Z2 was just an experimental prototype) shared a common architecture, with three main components: the memory, the processor, and the program. The memory (64 words for the Z1 and Z3, 1024 for the Z4) was used to store floating-point numbers. The processor operated on them using two internal registers, R1 and R2. The program commands were stored on a punched tape and they were read sequentially. The instructions were executed by the control unit, and the I/O console allowed the user to enter decimal numbers through a keyboard or read decimal results from lamps.

The clear separation between memory and processor, not present in machines such as the **ENIAC** or the **Harvard Mark I**, makes the Z1 and Z3 forerunners of what later was to be called **von Neumann architecture**. However, the program was not stored in memory, mainly because all machines had so few addressable memory locations. Zuse had thought of stored-program architectures but did not pursue the idea further.

The instruction set of the Z1 and Z3 consisted of the following operations: (1) addition, (2) subtraction, (3) multiplication, (4) division, (5) square root, (6) read a number from the keyboard to a register, (7) display the number in register R1, (8) store a number in memory, and (9) load a number from memory . Before doing any computation, numbers were read from the numerical **keyboard** to registers R1 and R2. Once the registers were loaded, addition, subtraction, multiplication, or division could be performed. The result was stored again in register R1. The time required by the Z3 for multiplication was 3 seconds. According to Zuse, the Z1 and Z3 were almost equally fast. The Z3 used 2000 relays—600 for the processor and 1400 for the memory.

Using the instruction set shown above, it is possible to process any arithmetical formula of the type used in engineering applications. However, the instruction set does not provide a conditional branching instruction, so it is relatively difficult, although not impossible, to perform more complex computations. Also, the punched tape can be bound to form a loop, so that iterative calculations are possible. Instructions were punched on the tape using an 8-bit code. Zuse's machines could be driven by the program stored in the tape or could be used with direct human input, much like modern pocket calculators.

The processor registers R1 and R2 could also be loaded with numbers stored in memory. Adding two numbers stored at memory locations X and Y and storing the result in memory location Z, consisted of the following sequence of operations: first, load R1 with the number stored in address X; then load R2 with the number stored in address Y; perform the addition; and finally, store the result in address Z. It is not difficult to see how more complex programs could be written for these machines.

Zuse avoided having to use many **logic gates** for the processor by using a control unit that worked as microsequencer, one for each command in the instruction set. A microsequencer consisted, in the case of the Z3, of a rotating arm that advanced one step in each cycle of the machine. A clock (a rotating motor) provided the clock cycles needed to synchronize the machine. In the case of the Z3, the operating frequency was about 5 hertz. At each cycle, the rotating arm in a microsequencer activated the circuits necessary for the operation at hand.

For example, in the case of multiplication, repeated addition and shifting of binary numbers were needed. The 18 partial operations needed were all started by a microsequencer with 18 contacts for the rotating arm. The microsequencer can be thought of as a hardwired program that implemented more complex instructions from atomic binary operations. Modifying the internal operation of the machine therefore consisted only in rewiring these microsequencers, without having to modify the rest of the processor. This resulted in a very efficient and flexible architecture, as was shown thereafter, when the instruction set of the Z3 was expanded for the Z4.

The Z4 would become the computer that Zuse had in mind from the beginning. It was built under contract for the German Airspace Research Office. The main difference between it and the Z3 was the larger memory (1024 addresses). However, using relays for the memory was out of the question, since this would have made the Z4 too bulky. Zuse developed a mechanical memory using the same types of components first used in the Z1. The final size of the larger memory was less than 1 cubic meter. The processor, on the other hand, was made of relays, and although it had the same basic structure as the processor of the Z3, it made more instructions available for the programmer. It was possible, for example, to multiply a number by 10, to shift a binary number, to calculate the reciprocal of a number, and so on.

All these extra instructions did not demand a fundamental change in the architecture of the machine, only that more microsequencing units be included in the machine. Multiplication by 10 was already effectively present as atomic operation in the Z3 (and was necessary to transform a number from the binary to the decimal notation) but was hidden from the programmer. In the Z4 many hidden instructions were made visible to the programmer, in this way increasing the usefulness of the machine. The main difference between the Z4 and Zuse's previous machines was the inclusion of conditional branching in the Z4, which enabled more sophisticated calculations to be performed.

After World War II the Z4 was moved from Berlin to Bavaria, where it stayed in a barn for almost four years. Eduard Stiefel from the Technical University of Zürich heard of the machine and after visiting Zuse, decided to rent the computer for his university. He asked Zuse to include conditional branching in the instruction set, a second punched tape for numbers, and to use a typewriter to print out the results. The machine was refurbished and conditional branching was added. The new instruction worked in the following way: When the contents of **register** R1was negative, the control unit skipped all following instructions until a special code ("start") was found in the punched tape. In this way it was possible to jump over sequences of instructions (i.e., those constituting the not-taken branch of a conditional instruction in the source code).

A detailed comparison of the numerical **algorithms** used by Zuse for the floating-point calculations in his machines with algorithms used in modern machines shows that they are almost identical. For division, Zuse used the approach now known as nonrestoring division. Square rooting was done using a similar method based on the division algorithm.

During the years 1941–49 Zuse developed plans for other machines and even applied for patents for some of them. He designed a logarithmic machine, in which numbers were stored as binary logarithms. He also designed the circuits for a device that he called a "logic machine," in which each memory word consisted of a single bit. The processor could only perform atomic logic operations (conjunction, disjunction, and negation) and a program was a sequence of such atomic operations. Perhaps the most surprising device he thought of was a machine that would be a front-end for the Z4: Formulas would be typed in a keyboard, using the standard mathematical notation, and the machine would transform them into assembler code for the Z4. This "Planfertigungsgerät," which was never completed, would have been a hardware **compiler**.

FURTHER READING

Rojas, Raúl, and Ulf Hashagen, eds. *The First Computers: History and Architectures.* Cambridge, Mass.: MIT Press, 1999.

Zuse, Konrad. *The Computer: My Life.* Berlin: Springer-Verlag, 1993.

—*Raúl Rojas*