Computer Science

Although it would be tempting to call Computer Science (CS) the science of building computers and making them work, there are actually several possible definitions. *Computing* Science would be a better name, since it stresses that this is the science that studies computation in all its possible manifestations, the algorithms thereof, and its possible applications. In other languages Computer Science has a more informative name: Informatik (in German), Informática (in Spanish), or Informatique (in French), stressing the fact that it is the science of automatic information processing.

Computer Science became a discipline only after the advent of computers. The first professional organization in the field was the Association for Computer Machinery (ACM) founded in 1947, only two years after the official presentation of the ENIAC. The first programmers and computer scientists came from the fields of Mathematics, Physics, and other branches of the natural sciences. Although the first CS departments were not officially started until the 1960s, some universities such as Harvard and Princeton started offering programming courses very early. Especially the Harvard Computation Laboratory, directed by Howard Aiken, could be considered as the precursor of later CS programs in the US.

It has been said that the term Computer Science was coined by Prof. George Forsythe from Stanford University in the 1960s. The first CS department was established at Purdue University in 1962, and the first person to receive a PhD in CS was Richard Wexelblatt at the University of Pennsylvania in 1965. Over many years, the CS departments at MIT, Stanford, Carnegie Mellon University, and Berkeley, have been consistently ranked as the best in the US.

Some authors consider CS a kind of applied branch of mathematics, others consider it a branch of engineering. Although the discussion might seem trivial, it goes to the heart of the matter. In the first case, CS is considered to be a science of abstract computable structures, i.e., computer scientists abstract a representation in terms of data structures from a practical problem, and develop algorithms to solve the problem. In the second case, CS as engineering, the emphasis is on the practical approach to the solution of computational problems.

There are four major subdivisions of CS and several sub-disciplines in each group. CS is divided into Theoretical, Technical, Practical, and Applied CS.

Theoretical CS deals with all theoretical aspects related to computational devices. *Computability theory*, for example, examines different hierarchies of machines and its relative computational power. A universal device, like the Turing Machine, is able to perform any computation that could be performed by a human provided with paper and pencil (and much patience). But there are alternatives, such as the Lambda Calculus or recursive functions, which are also capable of implementing universal computation. Whereas computability theory deals with the possibility of computing something, *complexity theory* deals with the efficiency with which the computation is done. If the computation can be performed in a number of steps which only increases polynomially with the size of the input, the computation is said to be feasible. If the number of steps increases exponentially, we are faced with a hard computational problem. Complexity theory gives guarantees for the efficiency of algorithms and reduces whole problem clases into other kinds of easier problems.
Other subdisciplines within theoretical CS are Automata Theory, Computational Geometry, Graph Theory, Formal Languages, and Circuit Complexity. Theoretical CS is in many cases

just another name for traditional topics which were previously investigated in mathematics departments.

Technical CS deals with the construction of computers, computers networks, and computing infrastructure in general. Building modern VLSI chips is no longer an art, but requires computer automated tools and rigorous standards. A chip containing 10 million transistors can only be designed using large CAD systems which profit from the latest results and methodologies developed at universities.

Practical CS is a more heterogeneous branch of CS, in which practical solutions to many different kinds of problems are developed. For example, writing large programs requires a special discipline studied in Software Engineering. This subfield of CS deals with measuring the complexity of programming projects, and explores how to subdivide them in modular portions which can be given to a small group or to a single programmer.

Database theory and practice is another area in which computer scientists are active. A database is any computerized repository of information, and there are several strategies to access this information. Relational databases, one alternative for the organization of the information, can be thought of as tables of data accessible through a number of primitive operations.

Another areas of practical CS are Artificial Intelligence (AI) and Robotics. AI, also called machine learning, deals with the means and methods to make computers simulate intelligent behavior. After much hype in the 1980s, the field has now consolidated and its results can be used in the emergent field of mobile autonomous robots.

Finally, applied CS covers research typically done in CS departments, but also outside of them or in strong collaboration with specialists from another disciplines. Some examples are Geographical Information Systems, Medical Informatics, Information Systems for Businesses or the emerging field of Bioinformatics, very related to theoretical CS.

In the US the number of PhDs awarded in CS reached around 1000 every year at the time of this writing. The yearly enrollment of undergraduates rose from 10,000 in 1995 to 20,000 in 1999, reflecting the increased need for CS professionals in industry. Even so, CS departments in the US were not able to provide all the specialists required by industry and many of them had to be recruited from abroad.

References

Mary Jane Irwin, Frank Friedman, *1998-1999 Taulbee Survey*, Computer Research News, March 2000.

Peter Freeman, William Aspray, *The Supply of Information Technology Workers in the US*, Computing Research Association, Washington, 1999.

Raúl Rojas