Turing Award Lecture "Computer Science: The Emergence of a Discipline" - Summary Report

SILPA C

Ph.D,IIITS

February 3, 2017

John E. Hopcroft received the Turing Award in 1986, and often recognized as the "Nobel Prize of computing for fundamental achievements in the design and analysis of algorithms and data structures". He had explained many of his experiences in the Turing award lecture "Computer Science: The Emergence of a Discipline". Article mainly focuses on rapid development in computer science. He explored how the computer science was invented in the engineering curriculum and the importance of computing in engineering, his own experiences in the field of computer science and growth in computing. He has worked with many other scientists like Huffman, Bakus and Naur , Chomsky, Pitts and Robert Tarjan who has contributed their work in Computer discipline.

John E. Hopcroft had published the following books :

- Introduction to Automata Theory, Languages and Computation
- The Design and Analysis of Computer Algorithms
- Formal languages and their relation to Automata
- Computing, Communication and the Information Age

The Evolution of computer Science and its discipline started earlier along with theory and practice. This discipline requires abstract and concrete modelling of real life problems. Computer Science has strong connectivity with other disciplines like mathematical solving, science predictions and domain specific knowledge.

Apart from this technology advancements has drawn vast challenges in collaborating the combined work of both academic institutions and industries in research perspective commitments. Fundamental analysis must be made from scratch considering algorithmic factoring, information analysis, circuit switching, logic design and theory of computation focusing more software and hardware design.

In 1994, changes in computing field took place focusing primarily work on digital computers with circuit designing and transistors. This laid out the need that computer scientist must not only focus on system in software or programming approach but feasible analysis lies in understanding the mathematical, science and try to incorporate them in efficient and optimal algorithmic design for good computations.

Acknowleding McCulloch and Pitts(19143), logical calculus paper describing events were series of electrical pulses viewed as strings of zeros and ones, which laid the development into the language of regular expressions. Another scientists, Rabin and Scott developed mathematical model which lead to the first state automation.

Computer science discipline is considered as formal science because of its root in mathematics. Computer programming is an important subfield in computer science due to its efficiency in theory and practical developments. Slowly, research focused more on structural programming, module design and verification rather than length of programming language with informal descriptions during 1960's. Acknowledging Naur and Backus for developing a notation for describing syntax for various programming languages, which was equal to Chhomsky.

In 1960's languages like Jovial, PL/I, Simula 67, Algol 68 and Snobol 50, these languages includes results of formal languages and automata theory with applications in parsing and compiling. More emphasis was on studying the abstract structures such as lambda calclus. Later in 1970's focus is more on developing programming languages with expressive power. Acknowledging Turing (1936) had introduced a turing machine was simple enough that any function computed by it was computable, and it is the foundation of modern computability theory.

The Complexity of algorithms is measured by number of steps during execution process but now we need to focus on correctness, concrete, ambiguous, finite, clarity and efficiency in terms of time and space complexity. Best and worst complexity metrics also depends on input values, problem size during execution process. Optimization of complex problem solving is also crucial during algorithmic design. Another challenge is classification and characterisation of complexities incorporating mathematical theories, NP- Completeness requires more depth understandings.

Later, focus is diverted towards algorithms and representation of input data using data structures. In 1960's more research is on algorithms and its comparative study on different programming languages in view of enhanced performance. Algorithm was measured by calculating rate of growth in computing time as function of problem size. Few questions arised like what is optimality in algorithm? What is intrinsic complexity of problem? How to achieve Asymptotic efficiency?

Acknowledging 1970's, John E. Hopcrotf focusing on working with integral part of computer science graph algorithms including planetary testing and graph connectivity. Algorithms with worst case optimal to constant factor also holds better than few existing algorithms. It attracted many of young researchers to establish data structures and algorithms as a subdiscipline. Slowly computer science achieved importance in cognitive sciences that introduced to work on various computational models with respect to interdisciplinary covering Artificial Intelligence with global structures. All these must support the semantics of language, abstraction and knowledge representations.

Computers have involved in all areas including agriculture, Medical, Education, Satellite communcation and business world, computers have made easy to think and live. In 1960's the computer science technology was broadened the scope from circuit design to computation. New ideas and methodologies will expand the scope of computer science. Existing programming languages requires higher level of skills to understand, but it must be reliable and user-friendly. Innovations are faster depends on programming environments.

More research is to be carried out in this field introducing and testing new programming models, interpreters that can efficiently handle much larger and complex systems and also develop science base in formal languages. Demand for computer scientists/researchers should come up with a national policy to support research activities in all aspects of computing and its advancements.