

COMPUTER LANGUAGE ADVANCES

DANIEL E. COOKE^{*,†,§}, MICHAEL GELFOND^{*}, and JOSEPH E. URBAN[‡]

**Computer Science Department, Texas Tech University,
P O Box 43104, Lubbock, TX 79409, USA*

†NASA Ames Research Center

*‡Computer Science and Engineering Department, Arizona State University
§E-mail: dcooke@coe.ttu.edu*

Computer language research began with the advent of the von Neumann architecture and proceeds to the present day, with promise for further advances in the future. This paper reviews the history of language research, and includes distinctions between experimental and theoretical research in the area, as well as a distinction between evolutionary and revolutionary advances. Major language paradigms are introduced and language design experience in the development of SequenceL is presented.

Keywords: Computer language, procedural/object-oriented, Lisp, Prolog, ordinary science, extraordinary science, abstraction, syntax, semantics.

1. Introduction

The beginning of modern computer science can be marked by Alan Turing's paper, which among other advances, showed that there are unsolvable problems [1]. Within the infinite set of solvable problems there is a set of technically feasible and problems that are not technically feasible to solve (e.g. problems that it would take a supercomputer 10,000 years to solve). Thus, from the beginning, computer science has been viewed as *the science of problem solving using computers*.

One category of computer science research is devoted to outwardly encroaching upon the set of intractable problems. Advances in computer hardware, for example, lead to more powerful computers that execute faster and have larger stores of memory. Advances in hardware are meant to result in improvements in the raw computing power of the devices that can be brought to bear in order to solve more complex problems. Additionally, research efforts focusing on complexity and algorithms are also concerned with outwardly encroaching upon the set of intractable problems.

In general, this first category of computer science research is devoted to making complicated problems more **technically feasible** to solve. Problems that are technically feasible are those for which existing algorithms can obtain an answer in some reasonable time using currently available computer technology. The second major category of computer science research has to do with making it **humanly feasible** to solve more complicated problems. Human feasibility has to do with the level of difficulty one faces in finding a solution to a problem.

Some of the efforts in the second category of research focus on the development of approaches that delegate some of the complexity of certain activities to easy-to-use tools, such as database management systems, operating systems, networking tools. Other approaches to the second category focus on problem representation. These areas of research include artificial intelligence and software engineering. Central to all efforts to improve upon the human's ability to solve more complicated problems is computer language research.

This chapter focuses on language design from a scientific point of view. To do so, a scientific backdrop will be developed. Then the notions of language design from a theoretical and experimental point of view are outlined. From there, the history of language design efforts are reviewed, first from the evolutionary and then from the revolutionary vantages. The chapter ends with a brief introduction of recent results obtained by one of the authors.

1.1. *Scientific endeavors*

In most sciences, there are two organized research fronts: theoretical and experimental. Experiment provides a wealth of scientific observations that eventually must be distilled into a compact representation. The distilled, compact representation is a theory — a formulation of the body of knowledge gained by experiment that concisely represents the larger body of knowledge. Once formulated, the theory is employed in the design of future empirical studies to see if the theory predicts the outcomes of the experiments. Future experiments may **confirm** the theory — leading to greater confidence in it — or they may **falsify** the theory — indicating that the theory may be incorrect. Results existing between confirmatory and falsification may indicate that the theory needs some modification towards improvement.

Based upon the interaction between the theoretical and experimental communities of the physical sciences, Kuhn, in his now classic text *The Structure of Scientific Revolutions*, identified two types of theoretical science [2]. The first category of theoretical research (called **ordinary science**) attempts to modify, extend, and/or refine theories so that they do a better job of explaining empirical results that are not explained by the unrefined theories. The second category (called **extraordinary science**) takes place when a radically new view (or theory) is proposed and it is seen that the theory explains empirical evidence, subsuming the old theory and all of its subsequent refinements. Extraordinary science results in paradigm shifts and in scientific revolutions.

2. Theoretical Language Research

A programming language serves as the central part of a theory about how one best approaches problem solving with the aid of computers. The language is central to the theory because it is the basis for finding and communicating problem solutions to machines or other people. A language is meant to facilitate one's ability to state and structure problem solutions. Possibly the most important aspect of the