**Sumary of “A Brief History Of Logic”**

This paper gives a succinct overview of the four major “ages” of logic throughout history, as well as a short introduction on the origin of the word “logic”.

Logic was an element of the *trivium*, a selection of fundamental curriculae which one had to master before before further study could commence. In an attempt to explain the term “logic”, some possible definitions are suggested. The original study of logic is attributed to the Sophists, who used it in an attempt to create an objective rule set for determining without a doubt who had won an argument. This is the essence of logic: helping to distinguish false from true, right from wrong.

The paper then continues with an enumeration of the ages of logic, the first one being Symbolic Logic, which reigned from 500 B.C. up until the 19th century. Logic has its roots in natural human language, and was used to determine the correctness of certain statements through definitions of the words used in a statement, and what could be inferred from those definitions. Unfortunately, natural language is very ambiguous in these definitions. A number of examples are given to illustrate this, such as the well-known “Liar’s Paradox”, the “Sophist’s Paradox” and the “Surprise Paradox”. Because of the ambiguity of natural language, attempts were made to transform the notion of logic into a symbolic language.

The second age was that of Algebraic Logic (mid-late 19th century), during which George Boole attempted to “investigate the fundamental laws of operations of the mind” by formulating logic in mathematical terms, using the likeliness between set union and intersection and numerical addition and multiplication. Charles Ludwig Dodgeson, who used Venn diagrams to reason about sets, and Ernst Schroëder, who foresaw the need for developing fast algorithms to solve these kinds of problems, are also prominent figures from this period. As the knowledge of symbolic logic grew, so did its importance in mathematics.

Then, the Age of Mathematical Logic (late 19th century – mid 20th century) emerged. Paradoxes began to appear in the more complex mathematical proofs, and logic was proposed to combat this trend. With this newfound rigor, many interesting questions were suddenly able to be reasoned about. Cantor investigated the notion of infinity, discovering that there is a whole hierarchy of infinities instead of just one, and David Hilbert even proposed a program to search for a single formalism upon which all mathematical truth rested. This was an utopian idea, however, with Russell’s Paradox and Kurt Gödel’s Incompleteness Theorems undermining the concept heavily. Still, major advances were made, such as the Principia Mathematica which formally proved the majority of the mathematical knowledge of that time, and existence of the Continuum Hypothesis and its relationship with mathematics. In the end, logic became an important part of mathematics, but not its foundation.

After this, logic found a new home in computer science when electronic computers started to emerge. Computer science uses formal systems and -languages to create new systems for use in computers. Modern logic is a cornerstone in many fundamental technologies of computer science, such as Boolean circuits, NP-completeness, programming language semantics, AI, program security and design validation & verification.

The paper concludes with the observation that, as our knowledge of logic and computer science expands, new relations between both continue to be discovered. Because of this, logic is sometimes described as “the calculus of computer science”.

**Another application of logic in computer science**

State Space Search is a very interesting branch of AI that explores a starting configuration in search of a (given) solution**.** The set of configurations (or “states”) can be pictured as a graph. Children of a node are legal operations on the parent state. Operations are actions one is allowed to do in a configuration to gain a new, different configuration. It is a very good way to solve “hard” problems such as chess (IBM’s Deep Blue II even beat Garri Kasparov in 1997), path-finding problems (such as those found in GPS-aided car navigation systems) and approximation algorithms to NP-complete problems. State space search relies on the concept of state space, which is a discrete dynamical system. Dynamical systems, in turn, are mathematical formalizations of rules that rely heavily on set theory. State Space Search itself can even be used to solve expressions in predicate calculus (<http://www.cs.trincoll.edu/~ram/cpsc352/notes/search.html>). State Space Search is also used in some algorithms for model checking, which in turn is used for automated theorem proving (used in design validation & verification of microchips).