Week2 Summary

General idea of this paper:

This article invents and introduces important features of LISP programming language including defining functions recursively, S-expressions and S-functions and how it works to realize these features. The core idea of LISP is function programming. He showed that with a few simple operators and a notation for functions, one can build a Turing-complete language for algorithms.

After that, the author also describes the representation of S-expressions in the memory of IBM 704. At last, the author summarized symbolic computations and gave a recursive function interpretation of flow charts.

Details about the paper:

The author first introduces functions and function definitions including partial functions, propositional expressions and predicates, conditional expressions, recursive function definitions, functions and forms and expressions for recursive functions. The author explains how to express these expressions with a function programming perspective.

In the middle part, the author talks about how to realize recursive functions of symbolic expressions. The author also provides a list of functions which can be used for writing out new functions to program. The article also introduces how lambda expressions works here, how a function can be used as an argument and so on. After that, John introduced the main aims of LISP language and its major data structures, including association lists, free-storage lists, representation of S-functions by programs.

At last, John mentioned another formalism for functions of symbolic expressions and compared linear LISP with LISP. He also illustrates LISP logic with a flowchart with single entrance and a single exit.

My Personal Thoughts:

LISP is very important function programming language and it is widely used in artificial intelligence technology these days. Function programming has many advantages, it is clear to read since we can figure out a clear pipeline of how the initial value change and develops. JavaScript is a typical functional programming language. Many other modern programming languages like Java, C++ are also absorbing this kind of programming style. Java imported the LAMBDA expressions in Java 8, C++ imported the LAMBDA in C++ 11. By using LAMBDA expression, we can also treat functions as objects, this is powerful tool for programming and I think I should learn more things about functional programming languages to fully take advantage of the convenience of these new features.