

FARROW83

Farrow,R.;  
Attribute Grammars and Data-Flow Languages;  
SIGPLAN'83 ACM (June 1983).

This article is a demonstration of the similarity between attribute grammars and data-flow languages. Attribute grammars were proposed by Knuth as a means of specifying programming language semantics. They are context-free grammars with attributes and semantic functions. The semantic functions specify values to be associated with attributes of a symbol in a production.

Data-flow languages are a sub-class of applicative languages. They have no concept of states or sequential execution. They have expressions -- powerful expressions. They lack variable reassignment and control-flow statements, consequently have no side-effects as well.

The underlying hypothesis that for any attribute grammar, there is a data-flow program that is an evaluator for the grammar. Semantic function design is equivalent to the problem of programming the data-flow language. Therefore, we can have attribute grammar-based translator-writing-systems that generate attribute evaluators written in some data-flow language. Also attribute grammars expose parallelism in compilation.

Farrow discusses various design issues related to data-flow languages such as abstract data-types to assist in intermediate code generation, dynamic binding of names to functions, logically supporting symbol tables, then, co-routines, recursion and looping. He enlightens the reader with the different problems that stem from the implementation of such facilities in a data-flow language.

Though the material is slightly complicated and detailed, this article is definitely food for thought in the area of compiler construction. The article is bound to effect the design of future data-flow languages, as well.