Title: Semantics of Programming Languages: A Tool-Oriented Approach*

Link: http://www.win.tue.nl/~mvdbrand/courses/seminar/0809/papers/p39-heering.pdf

What are the problems/research questions addressed by this article?

--->
The author states that The current situation in which semantics, languages, and tools are drifting steadily. The tool-oriented approach to semantics aims at making semantics definitions more useful and productive by generating as many language-based tools from them as possible. The goal is to produce semantically well- founded languages and tools. Ultimately, we envision the emergence of "Language Design Assistants" incorporating substantial amounts of semantic knowledge.

What are the existing solutions for this research question/problem?

--->
The semantics definition methods we are aware of are process algebras, and other methods that do not specifically address the semantics of programming languages, are not included. Dating back to the sixties, attribute grammars and denotational semantics are among the oldest methods, while abstract state machines (formerly called evolving algebras), coalgebra semantics, and program algebra are the latest additions to the field. Ironically, while attribute grammars are popular with tool builders, semanticists do not consider them a particularly inter- esting definition method.

What is the research method [s] they have used?

--->

A representative language development system (if any) for the semantics definition methods of above mentioned semantics. The Software Refinery, which has its origins in knowledge-based software .

What is their proposed solution ?

--->
The tool generation capabilities of the representative language development systems listed above. All of them can generate lexical scanners, parsers, and prettyprinters, many of them can produce syntax-directed editors, typecheckers, and interpreters,

and a few can produce various kinds of software renovation tools. To this end, they support one or more specification formalisms, but these differ in generality and application domain.

What are three future directions from this article?

-->
Action semantics.Plans (no longer pursued) for the Language Development Laboratory included a library of reusable language constructs, a knowledge base containing knowledge of languages and their com- pilers/interpreters, and a tool for language design.

Concepts that you learnt from this paper?

--->
parameterization and transformation .
Program slicer.
Origin tracker.
Call graph extractor.