

Putting Syntax to Work, for Real

1 Call-by-Name, Call-by-Value, and the λ -calculus

```
@article{gdp:cbn-cbv,  
  author={Plotkin, G. D.},  
  title={Call-by-Name, Call-by-Value, and the  $\lambda$ -calculus},  
  journal={Theoretical Computer Science},  
  pages={125--159},  
  year=1975  
}
```

Summary: Plotkin investigates the relation between ISWIM and the λ -calculus. In particular, he aims to determine what variant of the λ -calculus provides an equational theory that is consistent with the behavior of ISWIM programs. He establishes that the call-by-value λ -calculus models faithfully the meaning of ISWIM programs as evaluated by an SECD machine and that the equational theory of that version of the λ -calculus implies contextual equivalence for ISWIM. In addition, he shows that the call-by-name λ -calculus is a similarly appropriate model for a call-by-name version of ISWIM. Finally, he studies the relation between call-by-value and call-by-name. In particular, he constructs two simulations between a call-by-value and a call-by-name language and vice versa respectively, and shows that the simulations preserve equational reasoning but not contextual equivalence.

Evaluation: This is a seminal paper. It is the starting point of operational semantics and demonstrates that they are an extremely useful method to express the meaning of programs and prove properties of programming languages. Thus it kick-started a thread of work that established operational semantics as the main tool for studying programming languages. Furthermore, the limitations of the framing of the problem (ISWIM without mutation and control operators) and the negative results (the two simulations between the call-by-name and call-by-value languages and vice versa respectively do not preserve contextual equivalence) lead to a series of further research questions.