

# Automatically Constructing a Type System from the Small-Step Semantics

**Ștefan Ciobâcă** and Vlad Andrei Tudose

Faculty of Computer Science  
Alexandru Ioan Cuza University, Iași, Romania  
`stefan.ciobaca@info.uaic.ro`

May 31, 2017

# Motivation

1. design interpreter/compiler starting from (operational) semantics
2. design a type system for the language
3. prove the type system sound w.r.t. the operational semantics
4. (A) implement the type system as a type checker/type inferencer
5. (B) implement the operational semantics as an interpreter/compiler
6. Issue: no link between (A) and (B) – poor engineering.

# Motivation: state-of-the-art in type systems for programming languages

## 1. designs

# Conclusions and Future Work

1. From an engineering point of view, it is not ok to separate the interpreter and the type checker;

# Conclusions and Future Work

1. From an engineering point of view, it is not ok to separate the interpreter and the type checker;
2. It is possible to automatically construct a type system from the operational semantics;

# Conclusions and Future Work

1. From an engineering point of view, it is not ok to separate the interpreter and the type checker;
2. It is possible to automatically construct a type system from the operational semantics;
3. Question to the audience: please tell me about related work.