

# Compiler Construction

## Generating 1.21 GigaLOCs

Flip van Spaendonck & Lars Kuijpers

May 14, 2018



- Bottom-up parser
- Separate Lexer and Parser
- Written in Java



- Maps tokens to token traces
- Combines all token traces into a tree



- Expressions can get a special identifier
- Expressions with special identifiers are transformed in the tree
- We identify the following expressions:
  - Variable declaration
  - Variable assignments
  - Function declaration
  - Function calls
  - If-Else statements
  - While statements
  - Return statements



## Monomorphic Type Checking

- Check used IDs are well-defined
- Check basic function types
- Check defined function types
- Check boolean conditions (if and while)



- Finalizing tree structure
  - Made the type checking much easier

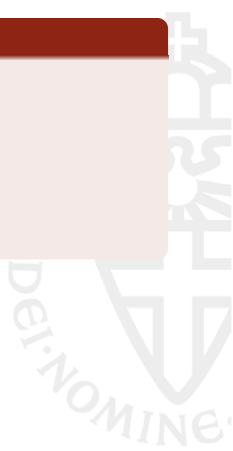


- Structure makes it easier to add extensions
- If you keep everything well-typed, currying works



## Fun Metrics

<b>LOC</b>	3616
<b>Word count</b>	9684
<b>Character count</b>	98470
<b>Sanity lost</b>	Even more
<b>Experience</b>	Priceless





# Questions ?

