

Dependently Typed Languages in Statix

Jonathan Brouwer Jesper Cockx Aron Zwaan

Delft University of Technology, The Netherlands

March 13, 2023

Background: What are Dependent Types?

- Types may depend on values!

Example

```
concat : (A: Set) -> (n : Nat) -> Vec A n -> Vec A n  
        -> Vec A (n + n)
```

- Curry-Howard correspondence

Research Question

How well Statix is fit for the task of defining a dependently-typed language.

Why is this important?

From the perspective of Spoofax research

Developing a language with a complex type system tests the boundaries of what Spoofax can do.

From the perspective of Dependent Types research

A rapid prototyping platform.

Calculus of Constructions

A lambda calculus with dependent types.

Example 1

```
(\v: Type. v) T
```

Example 2

```
let f = \T: Type. \x: T. x;  
f (T: Type -> Type) (\y: Type. y)
```

Type Checking

Type checking relation

`typeOfExpr : scope * Expr -> Expr`

How do we use scopes?

content...

Extra contributions

- 1 Implemented Inference
- 2 Implemented Inductive Data Types
- 3 Implemented Universes
- 4 Interpreter
- 5 Compiler to Clojure
- 6 Comparison with implementation in Haskell
- 7 Comparison with implementation in LambdaPi
- 8 Evaluation of Spoofax