

# EU TYPES WG 3: Types in Programming

Moderator: Andreas Abel

`andreas.abel@gu.se`

(Email me: include "COST WG3" in subject)

TYPES Conference 2018

Braga

18 June 2018

# Tasks

- 1 The design, study and deployment in a concrete programming environment of type theories that capture other properties **beyond functional correctness**, for example, *resource usage*, matching *communications*, *secure multi-party* computation, and *modularity*. (Related to WG1)
- 2 Design and deployment of **new strongly typed programming languages**, update of existing languages with refined type systems based on the findings of this work group.
- 3 Automated reasoning tools and proof assistants, including **type inference** (possibly also for untyped code), **program synthesis**, and **matching** (for example to retrieve a piece of code from a library that should satisfy a given specification).

## Some research topics of WG 3 (from 2016)

- Dependently typed programming
- Linear dependent types and session types
- Parametricity
- Other modalities (variance, irrelevance)
- Complexity analysis using types
- Type systems for security/privacy
- Type-safe meta- and generic programming
- Compiling dependent types
- Termination checking
- Universe polymorphism
- Partiality inside type theory
- Liquid/refinement typing
- Reactive programming

# COST Working Group Workshop (Nijmegen, Jan 2018)

- Joint workshop of COST WGs 1–4, organized by Herman
- WG3 talks:
  - *Combles* (Ralf Hinze)
  - *Intensional and univalent aspects of rule-based programming* (Ralph Matthes)
  - *Defunctionalisation as Modular Closure Conversion* (Ulrich Schöpp)
  - *Degrees of Relatedness* (Andreas Nuyts)
  - *Calculating correct programs* (Wouter Swierstra)
  - *Normalization by Evaluation for Sized Dependent Types* (Andreas Abel)

## Short Term Scientific Missions in WG3

Simon Castellan	Imp C London	Jorge Perez	U Groningen
<i>Probabilistic Session Types and their Semantics</i>			
Nobuko Yoshida	Imp C London	Jorge Perez	U Groningen
<i>Probabilistic Session Types and their Semantics</i>			
Jesper Cockx	U Gothenburg	Nicolas Tabareau	Nantes
<i>Extending definitional proof irrelevance with inductive datatypes</i>			
Lau Skorstengaard	Aarhus U	Dominique Devriese	KU Leuven
<i>Fully-abstract compilation between capability machines</i>			
Jurriaan Rot	U Nijmegen	Stefan Milius	FAU Erlangen
<i>Rational Causality</i>			
Atanas Hristov	U Ohrid	Anastasios Panagiotis	U Thessaloniki
<i>Type-based parallel programming</i>			
Henning Basold	ENS Lyon	Herman Geuvers	U Nijmegen
<i>Semantics and Extensions of Sized Inductive-Coinductive Types</i>			
Andreas Abel	U Gothenburg	Herman Geuvers	U Nijmegen
<i>Predicative semantics for sized dependent types</i>			

## Challenges and focus topics

- The art of dependently-typed programming (Conor McBride)
- Impact a greater audience
- Monadic dependently typed programming:  
Implement a (d.t.) programming language with well-scoped syntax!
- Can we compile better with dependent types? (Idris)
- Dependently-typed meta-programming
- Advance and utilize modal/quantitative type theories  $x :^m A \vdash t : B$
- ...

## Events

- FSCD 2018, 9–12 July 2018 and workshops PARIS, MSFP 7-8 July 2018
- ICFP 2018 and TyDe, 23–29 Sep 2018
- POPL 2019 (11 Jul) and CPP 2019 (? Oct), 13–19 Jan 2019 in Lisbon
- POPL 2019 Workshops!?! (30 June)
- Lambda Days 2019, Krakow, 21-22 Feb 2019 (500 participants)

# Old Slides

## Working Group 3: Types for programming

Type-based programming increases the security, reliability, resilience and reusability of software as evidenced by the increasingly popularity of strongly typed languages and the widespread use of types in applied programming language research.

The notion of type-based programming includes

- 1 programming languages that provide strong type systems that are either checked statically (Java, Haskell, Agda, ML) or dynamically (Scheme)
- 2 and to a limited extent also untyped languages (JavaScript, Tcl, PHP) that can only be analysed by third party tools,
- 3 as well as software engineering practises that use type theory.

# Activities, Milestones, and Major Deliverables

## Activities:

- 1 In year 1: set up the Working Group; have a joint meeting; define its research agenda.
- 2 In years 2, 3, 4: joint meeting of the WG; report on the progress in light of the research agenda; discuss and adapt the research agenda if needed.

## Milestones and Major Deliverables:

- 1 Minutes of the WG meeting
- 2 List of output of the past year
- 3 Research agenda

## What we want to do:

- Gather ongoing research efforts: topics, groups, papers
- Stimulate research and collaboration through
  - 1 WG 3 meetings (we can invite speakers)
  - 2 STSMs
- Document state-of-the-art and research agenda for EU