



A core language for optimised compilation of algebraic effect handlers

Axel Faes
Tom Schrijvers
KULeuven

What did I do?



Algebraic effect handlers

Exception handlers on steroids

BUT

Implementations have runtime penalty

```
effect Decide : unit -> bool;;
```

```
let choose_all = handler  
  | #Decide () k -> k true @ k false  
  | val x -> [x];;
```

```
with choose_all handle (if #Decide () then 10 else 20)  
(* Output: [10; 20] *)
```

OPTIMIZE



OPTIMIZE



ALL THE CODE?

memegenerator.net

New core language

Explicitly typed calculus with
row-based effects

Designed a type system

typing contexts $\Gamma ::= \epsilon \mid \Gamma, x : A$		
Expressions		
$\frac{\text{VAL}}{\Gamma \vdash v : A}$	$\frac{\text{VAR}}{(x : A) \in \Gamma} \Gamma \vdash x : A$	$\frac{\text{CONST}}{(k : A) \in \Sigma} \Gamma \vdash k : A$
$\frac{\text{FUN}}{\Gamma \vdash \lambda(x : A).c : A \rightarrow \underline{C}} \Gamma, x : A \vdash c : \underline{C}$	$\frac{\text{TYPE ABSTRACTION}}{\Gamma \vdash \Lambda \alpha. c : \forall \alpha. \underline{C}} \Gamma, \alpha \vdash c : \underline{C}$	
HAND		
$\frac{\begin{array}{l} C = A ! \{Op_i ; R\} \\ D = B ! \{Op_i ; R\} \quad (Op_i : A_{Op} \rightarrow B_{Op}) \in \Sigma \\ h = \{\text{return } x \mapsto c_r, [Op \ y \ k \mapsto c_{Op}]_{Op \in O}\} \\ \Gamma, x : A_{Op} \vdash c_r : \underline{D} \\ \Gamma, y : A_{Op}, k : B_{Op} \rightarrow \underline{D} \vdash c_{Op} : \underline{D} \end{array}}{\Gamma \vdash h : \underline{C} \Rightarrow \underline{D}}$		

Implementation

Implemented in Eff

Written in OCaml

Rewrote the type inference engine

```
124 and type_expr st {Untyped.term=expr; Untyped.location=loc} = type_plain_e
125
126 (* Type a plain expression *)
127 and type_plain_expr st loc = function
128 | Untyped.Var x ->
129   let ty_sch, st = get_var_scheme_env ~loc st x in
130   Ctor.var ~loc x ty_sch, st
131 | Untyped.Const const ->
132   Ctor.const ~loc const, st
133 | Untyped.Tuple es ->
134   let els = List.map (fun (e, _) -> e) (List.map (type_expr st) es) in
135   Ctor.tuple ~loc els, st
136 | Untyped.Record lst ->
137   let lst = List.map (fun (f, (e, _)) -> (f, e)) (Common.assoc_map (typ
138   Ctor.record ~loc lst, st
139 | Untyped.Variant (lbl, e) ->
140   let exp = Common.option_map (fun (e, _) -> e) (Common.option_map (typ
141   Ctor.variant ~loc (lbl, exp), st
142 | Untyped.Lambda (p, c) ->
143   let pat = type_pattern st p in
144   let comp, st = type_comp st c in
145   Ctor.lambda ~loc pat comp, st
146 | Untyped.Effect eff ->
147   let eff = infer_effect ~loc st eff in
```

Talk at main conference

[illegible]



Reflection

Choosing the honours project

Do I explore multiple interests?

Focus on a single aspect (area of research)



Time Management

Semester vs Summer

Deadlines

Deal with Summer feeling





ICFP 2017

First big conference

Able to network with a lot of people

Attend talks about different topics



PLMW

Specifically for beginning researchers

Talks about what (not) to do

Talks about writing papers

PLMW 2017 Programming Languages Mentoring Workshop 2017

[About](#)[Program](#)

Sun 3 Sep

09:00 - 09:10: PLMW 2017 - Welcome at L1

Chair(s): [Brigitte Pientka](#)

09:00 - 09:10

☆ Welcome

Talk

Brigitte Pientka, Neelakantan R. Krishnaswami, Dan Licata

09:10 - 10:00: PLMW 2017 - Keynote at L1

Chair(s): [Neelakantan R. Krishnaswami](#)

09:10 - 10:00

☆ Keynote

Talk

Chris Martens

10:30 - 11:30: PLMW 2017 - Session 1 at L1

10:30 - 11:00

☆ A Few Frank Remarks

Talk

Conor McBride

11:00 - 11:30

☆ Compositional Compiler Correctness

Talk

Amal Ahmed

[File Attached](#)

12:00 - 12:30: PLMW 2017 - Session 2 at L1

12:00 - 12:30

☆ Not How To Do Your PhD

Talk

Gabriel Scherer

[File Attached](#)

14:00 - 15:00: PLMW 2017 - Session 3 at L1

14:00 - 14:30

☆ Gradual Typing

Talk

Ronald Garcia

[File Attached](#)

14:30 - 15:00

☆ Scala: Types in Theory & Practice

Talk

Nada Amin

[File Attached](#)

15:30 - 16:30: PLMW 2017 - Session 4 at L1

15:30 - 16:30

☆ How to Write Papers and Give Talks That People Can Follow

Talk

Deek Dreyer

[Media Attached](#)

16:50 - 17:40: PLMW 2017 - Session 5 at L1

16:50 - 17:40

☆ Panel Discussion: Careers in Programming Languages

Talk

Sam Staton, Richard A. Eisenberg, Andreas Rossberg, Daan Leijen, Amal Ahmed

Student Research Competition

Practice how to explain work to an audience

Highlight: give talk at main conference

HONOURS STUDENT AXEL FAES WINS BRONZE MODEL IN ACM SIGPLAN

Honours student Axel Faes wins the bronze medal in the undergraduate category of the ACM SIGPLAN Student Research Competition held at the International Conference on Functional Programming in Oxford, UK for his work "Towards a core language with row-based effects for optimised compilation" supervised by prof. Tom Schrijvers.





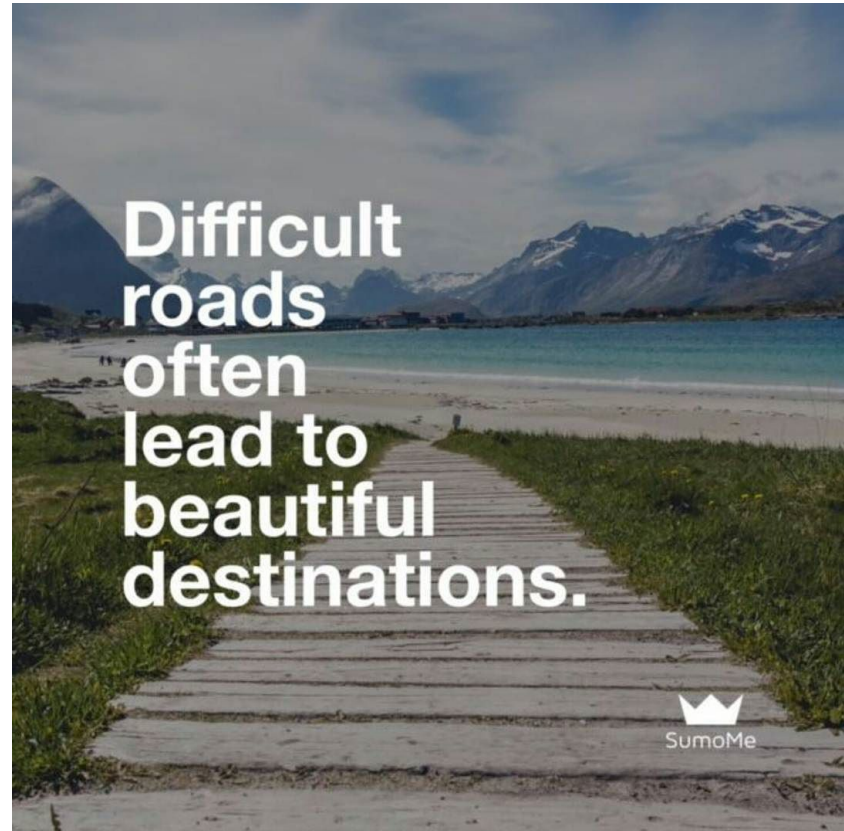
Looking back

Scope of the work

Theoretical

Implementation

Grown as a junior researcher



Conclusion
