Honoursprogramme: Research track, option A (9 ECTS)
Efficient Compilation of Algebraic Effects and Handlers in Eff

Axel Faes

prof. dr. ir. Tom Schrijvers
Amr Hany Shehata Saleh

Master in de ingenieurswetenschappen: computerwetenschappen (fase 1)
Specialisatie: Artificiel Intelligence

KULeuven

September 2016 - March 2017

Table of Contents

Overview

- Part of C1 project [1]
- Eff programming language [2]
- Type-&-effect system
- Compile to OCaml
- No optimizations $=>$ slow execution [3]

Steps

- Literature study
- Optimizations
- Benchmarks
- ~~Formal proof~~

Eff: N-queens sample code

```
effect Decide : unit -> bool
effect Fail : unit -> empty

let choose_all = handler
  | val x -> [x]
  | #Decide _ k -> k true @ k
      false
  | #Fail _ _ -> []

let queens_all nb_of_queens =
  with choose_all handle queens
       nb_of_queens
```

Description

Literature study

- Papers about algebraic effect handlers [4] [5] [6] [7]
- Papers about Eff [8] [9]
- Getting familiar with compiler

Optimizations

- with h handle (c) [c is pure for h] => c
- with h handle (c1 >> c2) [c1 is pure for h] => c1 >> (with h handle (c2))
- with h handle (c1 >> c2) [c2 is pure for h] => with h' handle (c1) [h' = c2 >> h.value_clause]
- with h handle (c1 >> c2) => with h' handle (c1) [h' = with h handle c2]
- with h handle (let rec defs = ... in c) => let rec defs = ... in (with h handle c)

Description

Benchmarks

Eff benchmarks

- Loops
- N-queens
- Interpreter [10]
- Parser [11]

Other systems

- Handlers in Action [12] [13]
- Eff directly in OCaml [14] [13]
- Multicore OCaml [15]
- ~~Links~~ [16] [17] [18]

Method of testing

- Bytecode
- Binary

Description I

Results

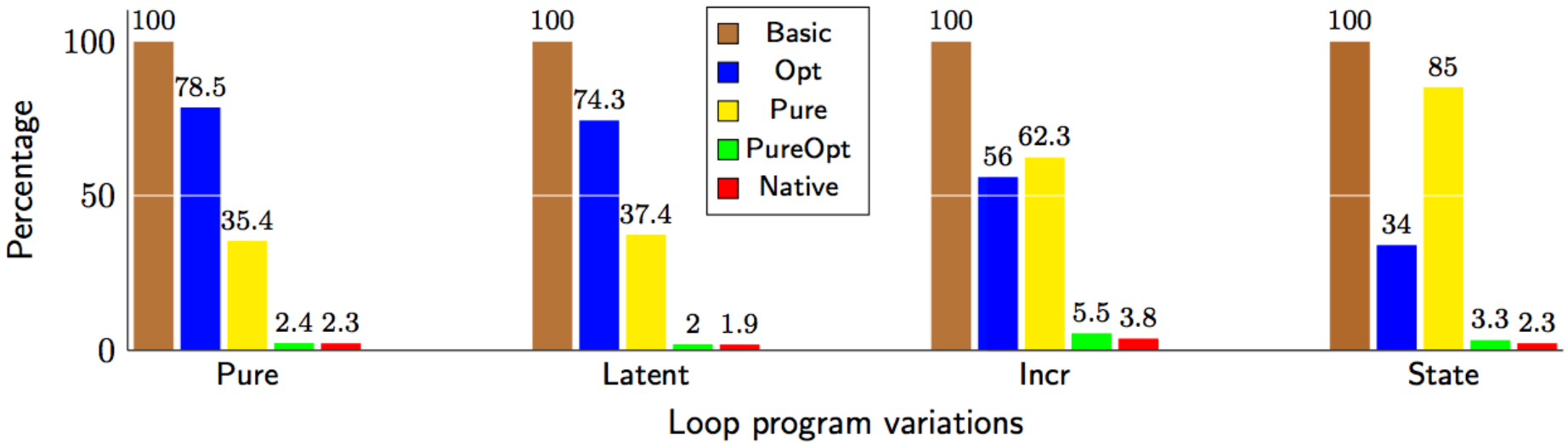

Figure: Relative run-times of Loops example [19]

Description II

Results

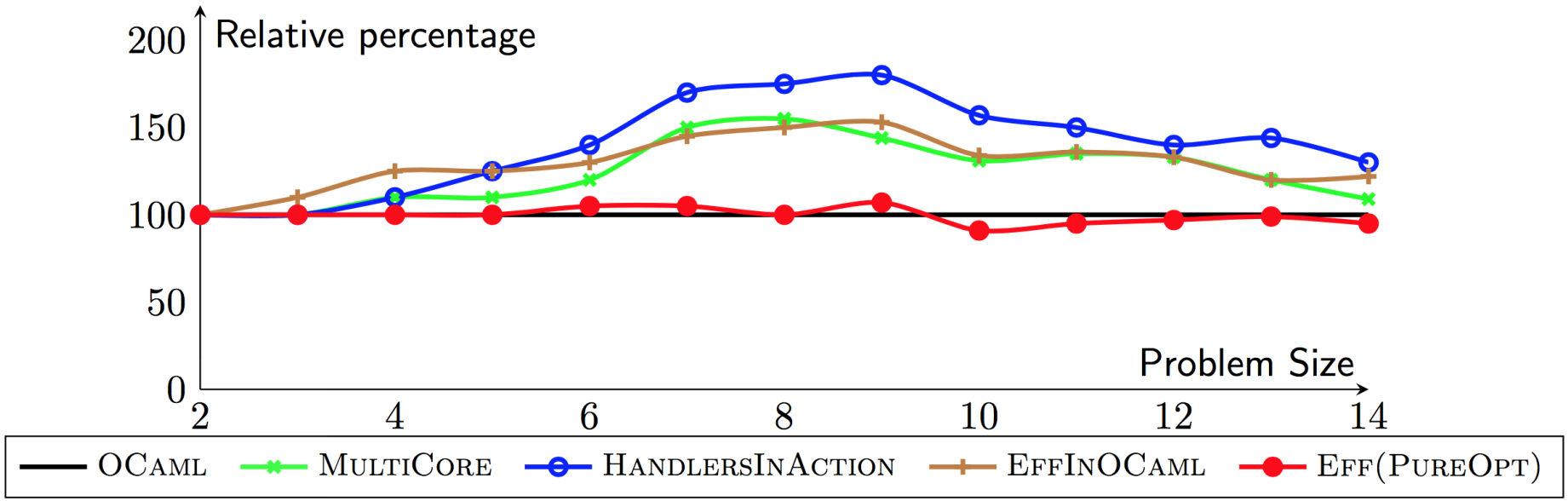

Figure: Results of running N-Queens for all solutions on multiple systems [19]

Reflection

General reflection

- Opportunity to perform research
- Challenging
- Done is one semester instead of throughout the year
- Effect of 70 ECTS + honours => huge workload
- Helped decide topic masterthesis
- Helped decide second honoursproject

Relation between honoursprogramme and degree

- Was familiar with functional programming
- Was familiar with compilers
- Formal Systems and their Applications (H04H8BE)
- Indirect synergy with other courses

Reflection

Oral Communication

- Weekly meetings
- English
- Shyness
- Difficult to find correct scientific terms
- Presentation for larger audience

Written communication

- Academic writing / paper
- Writing reports

Reflection

Theoretical and mathematical foundations of computer science

- Algebraic effect handlers
- Type systems

- Bigger thirst of knowledge
- Theoretical foundations

Asking for help

- Nothing wrong with asking for help
- Still needs improvement
- Assume too quickly that I understand

Reflection

Time management

- Big workload
- Learn how to manage time
- Still under/overestimate time required

Creating a hypothesis

- Improved during honoursproject
- but still a issue
- Difficulty to make concrete idea

Conclusion

- Accomplished more than expected
- Literature study took longer than expected
- Accomplished most steps from proposal
  - Literature study of Eff and existing optimizations
  - Designing new optimizations
  - implementation
  - evaluation through benchmarks
  - ~~formal proof of the optimizations~~
  - write parts paper
- Treated as researcher
- A big challenge/huge workload
- Found second honoursproject topic [18]
- Found masterthesis topic

References I

Tom Schrijvers. *Algebraic Effect Handlers: Harnessing the fundamental power of effects (3E160354)*. 2016. URL: https://www.kuleuven.be/onderzoek/portaal/#/projecten/3E160354.

Andrej Bauer and Matija Pretnar. *Eff*. 2016. URL: http://www.eff-lang.org/.

KC Sivaramakrishnan. *Eff directly in OCaml*. 2016. URL: https://github.com/kayceesrk/eff_delimcc_ocaml/blob/master/queens_results.pdf.

Niki Vazou and Daan Leijen. "From Monads to Effects and Back". In: *Practical Aspects of Declarative Languages: 18th International Symposium, PADL 2016, St. Petersburg, FL, USA, January 18-19, 2016. Proceedings*. Ed. by Marco Gavanelli and John Reppy. Cham: Springer International Publishing, 2016, pp. 169–186. ISBN: 978-3-319-28228-2. DOI: 10.1007/978-3-319-28228-2_11. URL: http://dx.doi.org/10.1007/978-3-319-28228-2_11.

Andrej Bauer and Matija Pretnar. "An Effect System for Algebraic Effects and Handlers". In: *Logical Methods in Computer Science* 10.4 (2014). DOI: 10.2168/LMCS-10(4:9)2014. URL: http://dx.doi.org/10.2168/LMCS-10(4:9)2014.

References II

Matija Pretnar. "Inferring Algebraic Effects". In: *Logical Methods in Computer Science* 10.3 (2014). DOI: 10.2168/LMCS-10(3:21)2014. URL: http://dx.doi.org/10.2168/LMCS-10(3:21)2014.

Gordon D. Plotkin and Matija Pretnar. "Handling Algebraic Effects". In: *Logical Methods in Computer Science* 9.4 (2013). DOI: 10.2168/LMCS-9(4:23)2013. URL: http://dx.doi.org/10.2168/LMCS-9(4:23)2013.

Matija Pretnar. "An Introduction to Algebraic Effects and Handlers. Invited tutorial paper". In: *Electr. Notes Theor. Comput. Sci.* 319 (2015), pp. 19–35. DOI: 10.1016/j.entcs.2015.12.003. URL: http://dx.doi.org/10.1016/j.entcs.2015.12.003.

Andrej Bauer and Matija Pretnar. "Programming with algebraic effects and handlers". In: *J. Log. Algebr. Meth. Program.* 84.1 (2015), pp. 108–123. DOI: 10.1016/j.jlamp.2014.02.001. URL: http://dx.doi.org/10.1016/j.jlamp.2014.02.001.

Sheng Liang, Paul Hudak, and Mark Jones. "Monad Transformers and Modular Interpreters". In: *Proceedings of 22nd ACM Symposium on Principles of Programming Languages*. New York: ACM Press, 1995, pp. 333–343.

References III

Nicolas Wu, Tom Schrijvers, and Ralf Hinze. "Effect Handlers in Scope". In: *SIGPLAN Not.* 49.12 (Sept. 2014), pp. 1–12. ISSN: 0362-1340. DOI: 10.1145/2775050.2633358. URL: http://doi.acm.org/10.1145/2775050.2633358.

Ohad Kammar, Sam Lindley, and Nicolas Oury. "Handlers in Action". In: *SIGPLAN Not.* 48.9 (Sept. 2013), pp. 145–158. ISSN: 0362-1340. DOI: 10.1145/2544174.2500590. URL: http://doi.acm.org/10.1145/2544174.2500590.

Oleg Kiselyov. *Implementations of delimited control in OCaml, Haskell, Scheme*. 2013. URL: http://okmij.org/ftp/continuations/implementations.html.

Oleg Kiselyov and KC Sivaramakrishnan. "Eff Directly in OCaml". In:

OCamlLabs. *MultiCore OCaml*. 2016. URL: https://github.com/ocamllabs/ocaml-multicore/wiki.

Links. *Links: Linking Theory to Practice for the Web*. 2016. URL: http://links-lang.org/.

References IV

Daniel Hillerström, Sam Lindley, and K Sivaramakrishnan. "Compiling Links effect handlers to the OCaml backend". In:

Daniel Hillerström and Sam Lindley. "Liberating Effects with Rows and Handlers". In: *Proceedings of the 1st International Workshop on Type-Driven Development*. TyDe 2016. Nara, Japan: ACM, 2016, pp. 15–27. ISBN: 978-1-4503-4435-7. DOI: 10.1145/2976022.2976033. URL: http://doi.acm.org/10.1145/2976022.2976033.

Matija Pretnar, Amr Hany Saleh, Axel Faes, and Tom Schrijvers. "Efficient Compilation of Algebraic Effects and Handlers". In: *ICFP 2017 (in review)* (2017).

The End