Circularity and λ-abstraction

Olivier Danvy, Aarhus University

Peter Thiemann, Universität Freiburg

Ian Zerny, Aarhus University

Landgoed Holthurnsche Hof IFL 2013

Once upon a time in the MidWest

Once upon a time in the MidWest

This is a talk about circular programs.

What is a circular program?

- A program seemingly in one pass

  that does two passes at once.

What is a circular program?

- A program seemingly in one pass

  that does two passes at once.
- Due to Bird in 1984: repmin.

What is a circular program?

- A program seemingly in one pass

  that does two passes at once.
- Due to Bird in 1984: repmin.
- Connected to circular attribute grammars.

What is a circular program?

- A program seemingly in one pass

  that does two passes at once.
- Due to Bird in 1984: repmin.
- Connected to circular attribute grammars.
- No simple operational intuition.

No simple operational intuition

Example: Dave Herman's research blog

"The Little Calculist" in July 2005, in Haskell.

- "strange-looking programs"
- "absolutely bizarre"
- "pretty weird"
- "trick"

No operational intuition is good!

"Reason equationally instead."

– Doaitse dixit.

But still

"I prefer call by value to call by name

because it is so more predictable."

– Mitchell Wand

For example

Reasoning about programs:

For a functional programmer,

a compositional complexity analysis

using structural induction

matches his/her operational intuition.

No operational intuition, really?

- Pettorossi in 1989 (using my words):

  1. accumulate a delimited continuation;
  2. and then apply it.

No operational intuition, really?

- Pettorossi in 1989 (using my words):
  1. accumulate a delimited continuation;
  2. and then apply it.
- But is this circular programming?

No operational intuition, really?

- Pettorossi in 1989 (using my words):
  1. accumulate a delimited continuation;
  2. and then apply it.
- But is this circular programming?
- For example, writing it back to direct style doesn't give a Bird-style circular program.

The situation

Bird-style circular programs

vs.

Pettorossi-style circular programs.

The situation

Bird-style circular programs

vs.

Pettorossi-style circular programs.

- Are they the same?

The situation

Bird-style circular programs

vs.

Pettorossi-style circular programs.

- Are they the same?
- Is the Emperor naked?

A brave attempt

- strictify a Bird-style circular program

  (with thunks and recursive bindings)

A brave attempt

- strictify a Bird-style circular program

  (with thunks and recursive bindings)

- promptly get lost

A second, more cautious attempt

- analyze the strictness of the circular program

A second, more cautious attempt

- analyze the strictness of the circular program
- strictify it with a smaller number of thunks

A second, more cautious attempt

- analyze the strictness of the circular program
- strictify it with a smaller number of thunks
- try to transform it towards Pettorossi style

The usual suspects (and their left inverses)

- closure conversion
- lambda-lifting
- defunctionalization
- CPS transformation

The usual suspects (and their left inverses)

- closure conversion
- lambda-lifting
- defunctionalization
- CPS transformation

...but it leads nowhere.

Several years pass...

"A problem worthy of attack

proves its worth by hitting back."

– Piet Hein

Success at last at AFP 2008

- lambda-lifting and lambda-dropping
- 7 easy steps
- applicable to several examples
- reversible

Stamp of success

- Richard Bird (another AFP lecturer) likes it.
- Towards a functional pearl.

A new player

- Ian Zerny

A new player

- Ian Zerny
- a warmup project

A new player

- Ian Zerny
- a warmup project
- more examples

A new player

- Ian Zerny
- a warmup project
- more examples
- but critical mass not reached:

  7 is a big number

A new player

- Ian Zerny
- a warmup project
- more examples
- but critical mass not reached:

  7 is a big number
- we start doing other things,

  and never come back

An opportunity

- Doaitse's Liber Amicorum

An opportunity

- Doaitse's Liber Amicorum
- Peter Thiemann

An opportunity

- Doaitse's Liber Amicorum
- Peter Thiemann
- he understands the 7 steps

  and agrees that they work

An opportunity

- Doaitse's Liber Amicorum
- Peter Thiemann
- he understands the 7 steps

  and agrees that they work
- he suggests a new example due to Knuth

An opportunity

- Doaitse's Liber Amicorum
- Peter Thiemann
- he understands the 7 steps

  and agrees that they work
- he suggests a new example due to Knuth
- we all rewrite the paper

Michelangelo to the rescue

Simplicity pays off

We were simplifying our presentation,

and at one point

the simplification suggested a simpler solution.

- In the strictified lazy programs, inductively, we add something to a dynamic unknown.

- In the strictified lazy programs, inductively,
  we add something to a dynamic unknown.
- Separation of concerns: pair the unknown
  with what we add to it,
  which we represent as a function.

- In the strictified lazy programs, inductively,
  we add something to a dynamic unknown.
- Separation of concerns: pair the unknown
  with what we add to it,
  which we represent as a function.
- The dynamic unknown is never used:
  it is a dead variable, so we kill it.

- In the strictified lazy programs, inductively,
  we add something to a dynamic unknown.
- Separation of concerns: pair the unknown
  with what we add to it,
  which we represent as a function.
- The dynamic unknown is never used:
  it is a dead variable, so we kill it.
- The result is in Pettorossi style!

- In the strictified lazy programs, inductively,
  we add something to a dynamic unknown.
- Separation of concerns: pair the unknown
  with what we add to it,
  which we represent as a function.
- The dynamic unknown is never used:
  it is a dead variable, so we kill it.
- The result is in Pettorossi style!
- And it applies to all our examples.

Plus

Also, our transformation is reversible.

Plus

Also, our transformation is reversible.

And this applies to all our examples.

Plus

Also, our transformation is reversible.

And this applies to all our examples.

See the paper in the pre-proceedings.

And since

- Derek Dreyer's visit

And since

- Derek Dreyer's visit
- cofix in Coq

And since

- Derek Dreyer's visit
- cofix in Coq
- our logical next step

Conclusion for now

- An operationalization of circular programs.

Conclusion for now

- An operationalization of circular programs.
- A conciliation of Bird and Pettorossi.

Conclusion for now

- An operationalization of circular programs.
- A conciliation of Bird and Pettorossi.
- An inter-transformation usable by others.

Conclusion for now

- An operationalization of circular programs.
- A conciliation of Bird and Pettorossi.
- An inter-transformation usable by others.
- New examples.

Conclusion for now

- An operationalization of circular programs.
- A conciliation of Bird and Pettorossi.
- An inter-transformation usable by others.
- New examples.
- The likelihood of a formalization with Coq.

Conclusion for now

- An operationalization of circular programs.
- A conciliation of Bird and Pettorossi.
- An inter-transformation usable by others.
- New examples.
- The likelihood of a formalization with Coq.

...and a 30-years-old mystery solved.

Credits

- Joint work with Peter Thiemann & Ian Zerny.
- Originally for Doaitse Swierstra's Festschrift.
- Presented at IFL 2013.