# Contextual Dispatch for Function Specialization

## Doctoral Research Days at FIT 2020

**Jan Ječmen**

**Faculty of Information Technology, Czech Technical University in Prague**

**Supervisor: Jan Vitek**

**November 27, 2020**

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

# Study info summary

Student: Jan Ječmen

Supervisor: Jan Vitek

Year of studies: 4

Dissertation thesis topic: Just-in-time Interprocedural Optimization for Dynamic Programming Languages

# Motivation

JIT compiler for

Have to deal with laziness, side-effects, reflection, and more

# Consider:

```
f <- function(n) {
    i <- 0
    while (i < n) {
        print(i)
        i <- i + 1
    }
}
```

# Consider:

```
f <- function(n) {
    i <- 0
    while (i < n) {
        print(i)
        i <- i + 1
    }
}

f(3)
## [1] 0
## [1] 1
## [1] 2
```

# Consider:

```
f <- function(n) {
    i <- 0
    while (i < n) {
        print(i)
        i <- i + 1
    }
}

f( { print("hi"); 3 } )
## [1] "hi"
## [1] 0
## [1] 1
## [1] 2
```

# Consider:

```
f <- function(n) {
    i <- 0
    while (i < n) {
        print(i)
        i <- i + 1
    }
}
evil <- function() { assign("i", 1, sys.frame(-1)) }
f( { evil(); 3 } )
## [1] 1
## [1] 2
```

# Contextual dispatch

```
f(1000)  ## -----> f_safe(1000)

f(evil())  ## -----> f_anything(evil())
```

# Context

Property of a program state
Examples: bool flags, types, shape, quantity

Efficiently comparable – contexts form lattices

```
                T
                |
            Arg0Eager
            /       \
      Arg0Eager     Arg0Eager
      Arg1Int       Arg1Real
```

# Context

For a callsite with a current context $C$ we can dispatch to any function version $[V_n, C_n]$ such that $C \leq C_n$

For example, the callsite

```
    f(1L, 3.14)  # -----> C = (int, real)
```

can dispatch to any of $(int, any)$, $(any, real)$, and $(any, any)$

Which is best? There are two smallest contexts...
Either pick one randomly
Or compile version for $(int, real)$!

# Performance

On benchmarks on average 1.7x vs. GNU R, 0.6x vs. FastR
CD improves 18/46 benchmarks

# Publications

1. Flückiger, Olivier and Chari, Guido and Yee, Ming-Ho and Ječmen, Jan and Hain, Jakob and Vitek, Jan. 2020. *Contextual Dispatch for Function Specialization* Proc. ACM Program. Lang. 4, OOPSLA, Article 220 (November 2020), 36 pages. https://doi.org/10.1145/3428288

# Thank you!

*github.com/reactorlabs/rir*

**Acknowledgements**
This work has received funding from the Office of Naval Research (ONR) award 503353, the National Science Foundation awards 1544542 and 1618732, the Czech Ministry of Education, Youth and Sports from the Czech Operational Programme Research, Development, and Education, under grant agreement No. CZ.02.1.01/0.0/0.0/15_003/0000421, and the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, under grant agreement No. 695412.