

# Typing R

Elie Canonici Merle

CNAM

July 11, 2019

Supervisors:

Pierre Courtieu CNAM

Christophe Genolini Zébrys

# Typing : Dynamic vs Static

Dynamic

At runtime

Static

Before runtime

# Typing : Dynamic vs Static

## Dynamic

At runtime

Flexibility

## Static

Before runtime

Safety

# Typing : Dynamic vs Static

## Dynamic

At runtime

Flexibility

Light syntax

## Static

Before runtime

Safety

Verbose syntax

# Typing : Dynamic vs Static

## Dynamic

At runtime

Flexibility

Light syntax

## Static

Before runtime

Safety

Verbose syntax

In a perfect world:

Flexibility

Safety

Light syntax

# The industry is slowly shifting

Javascript : TypeScript, Flow, Reason

Python : mypy

What about R?

# Some of R challenges

Lexical scoping with late binding

Lazy argument evaluation with potential side effects

Reflexive features of the language

Redefining the primitives of the language



Our approach

Keep R syntax

# Our approach

Keep R syntax

Type what we can type

# What can we borrow from the static world?

Type inference : keep syntax light

Effect system : capture some of R semantic

Gradual typing : usable out of the box

# Some examples

# Undefined identifiers

```
variable <- 42
```

```
variabl
```

# Undefined identifiers

```
variable <- 42
```

```
variabl
```

```
Error: object 'variabl' not found
```

# Undefined identifiers

```
variable <- 42
```

```
variabl      Unknown identifier 'variabl'
```

# Undefined identifiers

```
variable <- 42
```

```
variable
```



# Control Flow Structures

```
x <- scan(n=1)
```

```
if (x == 0) {  
  y <- 42  
} else {  
  z <- 42  
}
```

y

z

# Control Flow Structures

```
x <- scan(n=1)
```

```
if (x == 0) {  
  y <- 42  
} else {  
  z <- 42  
}
```

```
y
```

```
Error: object 'y' not found
```

```
z
```

```
Error: object 'z' not found
```

# Control Flow Structures

```
x <- scan(n=1)
```

```
if (x == 0) {
```

```
  y <- 42
```

Identifier 'y' may not exist

```
} else {
```

```
  z <- 42
```

Identifier 'z' may not exist

```
}
```

y

z

# Control Flow Structures

```
x <- scan(n=1)
```

```
y <- 0
```

```
z <- 0
```

```
if (x == 0) {
```

```
  y <- 42
```

```
} else {
```

```
  z <- 42
```

```
}
```

```
y
```

```
z
```

# Control Flow Structures

```
x <- scan()
```

```
for (i in x) {  
  z <- 42  
}
```

```
z
```

# Control Flow Structures

```
x <- scan()
```

```
for (i in x) {  
  z <- 42  
}
```

```
z
```

```
Error: object 'z' not found
```

# Control Flow Structures

```
x <- scan()
```

```
for (i in x) {  
  z <- 42  
}
```

Identifier 'z' may not exist

```
z
```

# Control Flow Structures

```
x <- scan()
```

```
z <- 0
```

```
for (i in x) {  
  z <- 42  
}
```

```
z
```



# Function Calls

```
f <- function(x = 1, y) y
```

```
f(z=1)
```

```
f()
```

# Function Calls

```
f <- function(x = 1, y) y
```

```
f(z=1)
```

```
Error in f(z = 1) : unused argument (z = 1)
```

```
f()
```

```
Error in f() : argument "y" is missing, with no default
```

# Function Calls

```
f <- function(x = 1, y) y
```

**f(z=1)**            function 'f' does not have a 'z' argument

**f()**                function 'f' requires a 'y' argument

# Function Calls

```
f <- function(x = 1, y) y
```

```
f(y=1)
```

```
f(1)
```

# Higher order

```
f <- function(g) {  
  g(x = 10)  
}
```

```
f(function(y) y)
```

# Higher order

```
f <- function(g) {  
  g(x = 10)  
}
```

```
f(function(y) y)
```

Error in g(x = 10) : unused argument (x = 10)

# Higher order

```
f <- function(g) {  
  g(x = 10)  
}
```

```
f(function(y) y)
```

function with a 'x' argument expected

# Higher order

```
f <- function(g) {  
  g(x = 10)  
}
```

```
f(function(x) x)
```



# Free variables and late binding

```
f <- function() x
```

```
f()
```

# Free variables and late binding

```
f <- function() x
```

```
f()
```

```
Error in f() : object 'x' not found
```

# Free variables and late binding

```
f <- function() x
```

f()

Unkown identifier 'x'

# Free variables and late binding

```
f <- function() x
```

```
x <- 42
```

```
f()
```

# Super assignments

```
f <- function (b) if(b) x <- 42 else 42
```

```
f(FALSE)
```

```
x
```

```
f(TRUE)
```

```
x
```

# Super assignments

```
f <- function (b) if(b) x <- 42 else 42
```

```
f(FALSE)
```

```
x
```

```
Error: object 'x' not found
```

```
f(TRUE)
```

```
x
```

# Super assignments

```
f <- function (b) if(b) x <- 42 else 42
```

f(FALSE)      Identifier 'x' may not exist

x              Unkown identifier 'x'

f(TRUE)        Identifier 'x' may not exist

x              Unkown identifier 'x'

# Super assignments

```
f <- function (b) if(b) x <- 42 else 42
```

```
x <- 0
```

```
f(FALSE)
```

```
x
```

```
f(TRUE)
```

```
x
```



## Ongoing and future work

Expand on what we can type

Optimizing compilation

Thank you for your attention