Debugging and expanding Meta Casanova

Louis van der Burg

June 26, 2016

Introduction

- Motive
- Research questions
- Results
- Conclusion
- Questions

Motive

- Video game industry
- Casanova
- Meta Casanova

Research question

How can the programming language Meta Casanova be improved for the user within the timeframe of the internship?

Sub research questions

- What is a good programming language to the user?
- What is MC and how does it work?
- How can the current syntax be improved to serve the user?
- How can the standard library be improved to serve the user?

What is a good programming language to the user?

Criteria:

- Read- & Writability
- Simplicity
- Definiteness
- Predictability
- Expressiveness
- Implementability
- Efficiency
- Custom Libraries
- Time
- Hackability
- Succinctness
- Redesign
- External Factors

Criteria are used as guidelines

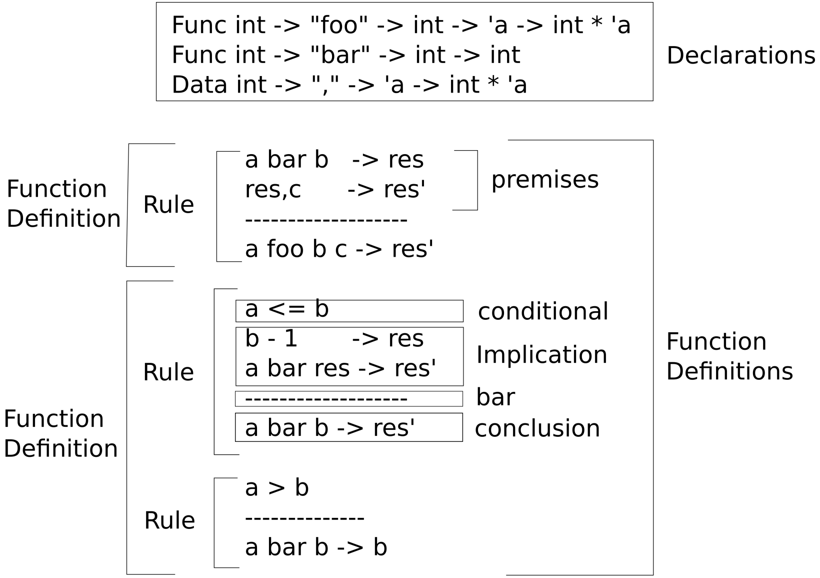
What is MC and how does it work?

Meta Casanova

- Functional
- Declarative
- Pure

What is MC and how does it work?

Basics: part one



```
Func int -> "foo" -> int -> 'a -> int * 'a
Func int -> "bar" -> int -> int
Data int -> "," -> 'a -> int * 'a
```

Declarations

Function Definition

Rule

```
a bar b   -> res
res,c     -> res'
-------------------
a foo b c -> res'
```

premises

Function Definition

Rule

```
a <= b
b - 1        -> res
a bar res -> res'
-------------------
a bar b -> res'
```

conditional

Implication

bar

conclusion

Rule

```
a > b
--------------
a bar b -> b
```

Function Definitions

What is MC and how does it work?

Basics: part two

- TypeFunc
- TypeAlias
- Module

How can the current syntax be improved to serve the user?

Expanding Modules

- Old

```
TypeFunc "expanding" => Module => Module
expanding M => M{
    ......
}
```

- New

```
TypeFunc "expanding" => Module => Module
expanding M => Module{
    inherit M
    ......
}
```

How can the current syntax be improved to serve the user?

Syntax additions

- Priority

```
Func "bar" -> Int  #> 12 R
TypeFunc "foo" => Float => 'a => 'b  #> 9
```

- .NET

```
import System

Func "dotNetTest" -> String
dotNetTest -> DateTime.Now.ToString()
```

How can the standard library be improved to serve the user?

- Prelude
- Number
- Record
- Monads

How can the standard library be improved to serve the user?

Prelude

```
TypeAlias "Then" => Type
Data "then" -> Then

TypeAlias "Else" => Type
Data "else" -> Else

Func "if" -> Boolean^System -> Then -> 'a ->
   ↪ Else -> 'a -> 'a
if True^builtin then f else g -> f
if False^builtin then f else g -> g
```

How can the standard library be improved to serve the user?

Number & Record

- Number
  - Generic
- Record
  - Compile time

How can the standard library be improved to serve the user?

Monads

- Monad
- Monad transformers
- TryableMonad

How can the standard library be improved to serve the user?

Implemented monads & monad transformers

- List
- Either
- Result
- State
- Id

Advantages of Meta Casanova

- Predictability
- Definiteness
- Succinctness
- Safety
- Choice of execution time

Conclusion

- Sub research questions
  - What is a good programming language to the user?
  - What is MC and how does it work?
  - How can the current syntax be improved to serve the user?
  - How can the standard library be improved to serve the user
- Main research question
  - How can the programming language Meta Casanova be improved for the user within the timeframe of the internship?
- Video game industry

Conclusion

Conclusion

Conclusion

Questions

Questions