Type System for Metaprogramming

Junyoung "Clare" Jang
https://Ailrun.github.io
Complogic Group

Metaprogramming

The art of generating, manipulating, and analyzing code
⟵ Compiler,Program Analyzer,Program generator,etc.

`box(2 + 3)` A code fragment for code `2 + 3`

`let box(C) = box(2 + 3) in C` Execution of `C` to get `5`

```
nth n =
  if n <= 0 then
    box(head xs)
  else
    let box(C) = nth (n - 1) in
    box(let xs = tail xs in C)
```

Untyped Metaprogram to get nth item in a list with no type/scope checking

Some text text text text text text text text text text text text text text text text text text text text text text text text text text text text text

Typed Metaprogramming

Some text text text text text text text text text text text text text text text text text text text text text text text text text text text text

```
nth : Int -> □ (Int list -> Int)
nth n =
  if n <= 0 then
    box(fun xs => head Int xs)
  else
    let box(C) = nth (n - 1) in
    box(fun xs => C (tail Int xs))
```

Typed Metaprogram to get nth item in an int list

Some text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text

Problem 1 — Well-typed Open Code Fragments

Some text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text

Problem 2 — Polymorphism

Some text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text

```
nth : (a' : Type) -> Int -> □ ('a list -> 'a)
nth a' n =
  if n <= 0 then
    box(fun xs => head a' xs)
  else
    let box(C) = nth a' (n - 1) in
    box(fun xs => C (tail a' xs))
```

Ill-typed Metaprogram to get nth item in a list

Some text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text

Problem 3 — Pattern Matching on Code Fragments

Some text

Modal Logic

Some text

Contextual Modality

Some text

Levels (Opt)

Some text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text

Levels in Polymorphism

Some text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text

Levels in Pattern Matching

Some text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text

References

Some text