Type Universes and Heaps?

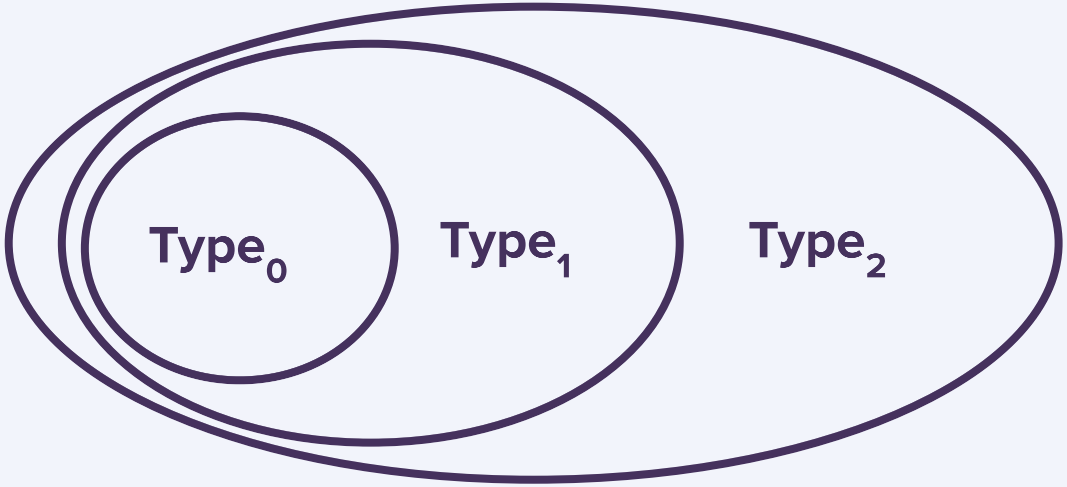
You Won't Believe What They Have in Common

Paulette Koronkevich
William J. Bowman

What’s a type universe?

We *looooove* reasoning about types themselves, of type **Type**.

But **Type : Type** is inconsistent, so we have $\mathbf{Type_0} : \mathbf{Type_1} : \mathbf{Type_2} : \dots$



… and heaps?

Simple Functional Language

+

Mutable References

=

Unrestricted Recursion

Simple Functional Language

+

Mutable References

+

Type Universe Hierarchy

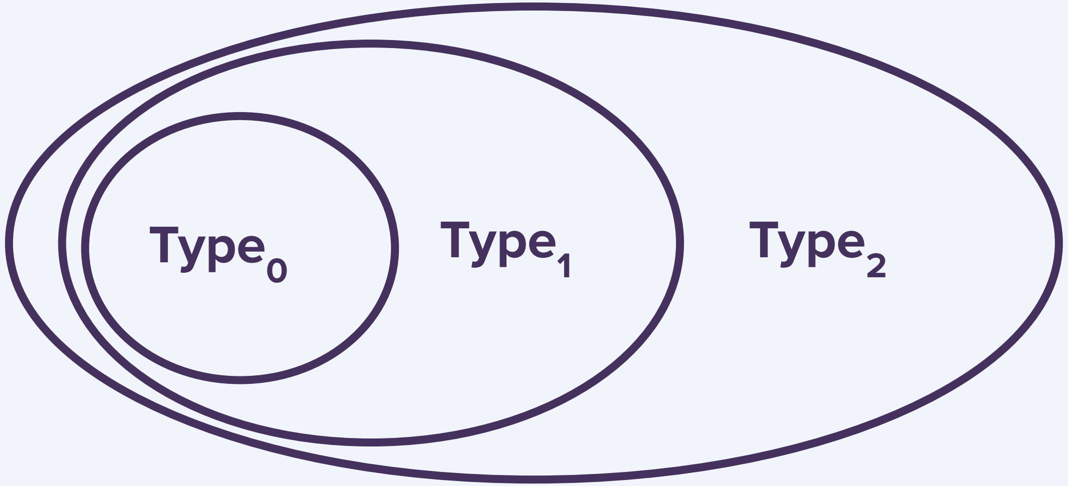
=

Higher Order References

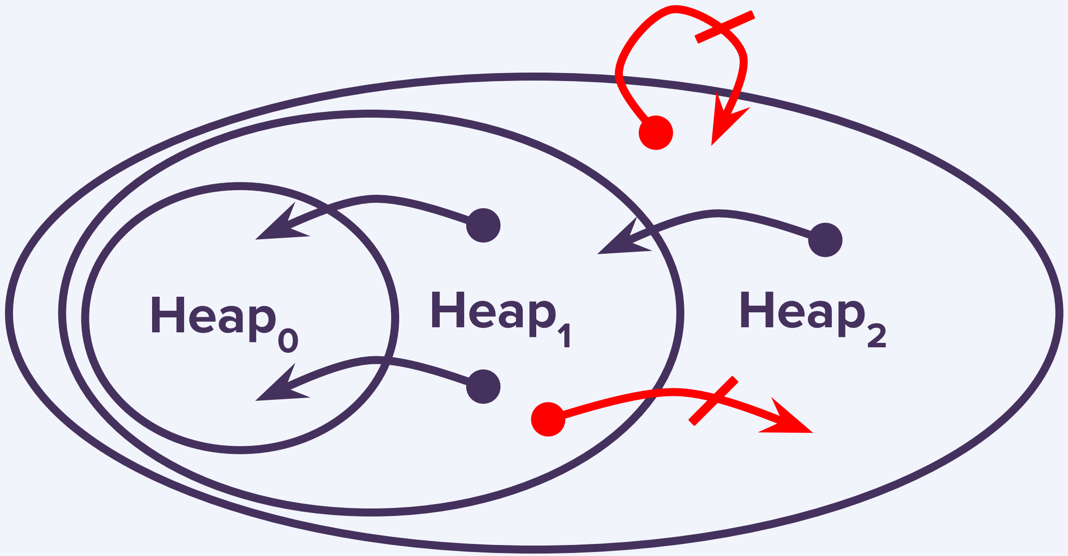
+

Termination!!

BUT HOW!?!?!



BUT HOW!?!?!



This type system is...

take that, Rust!

- Declarative and synta
- Simple, no ownership
- Terminating!

When you decide to use mutable references in Rust
Realizing you need to borrow and move the same variable
Trying to convince the compiler it's okay
Compiler: Error[E0505] cannot move out of tcp_stream because it is borrowed
imgflip.com

Lots of future directions

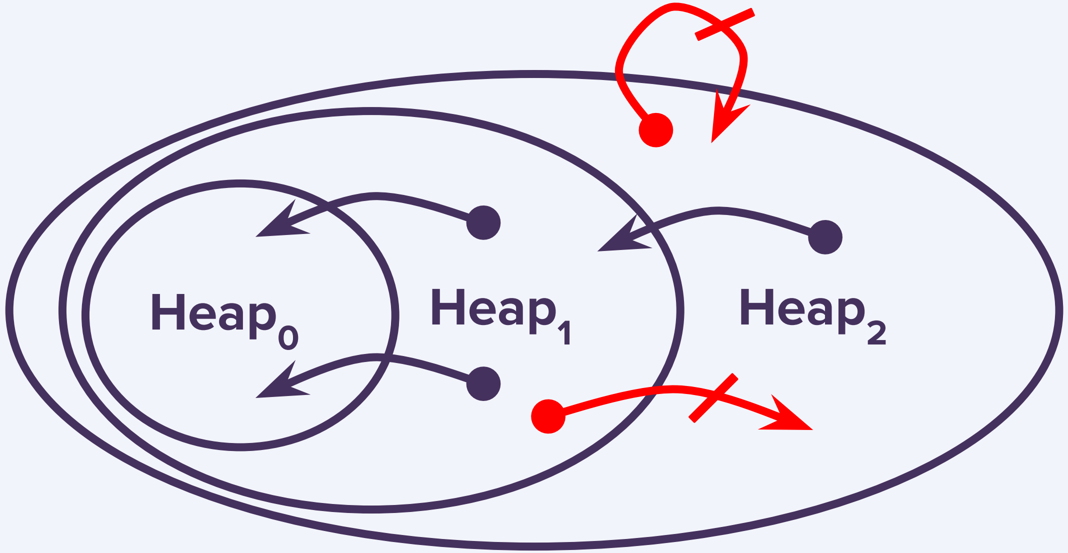
Lots of future directions

Injecting existing work from type universes

Like a region system (but worse), Type Universes ⇔ Regions

… and any you might have?

Thank you!


$Heap_0$
$Heap_1$
$Heap_2$