Type Annotations for Reference Types

Andreas Rossberg
Dfinity

Originally

ref.is_null : [anyref] → []

canonical (“principal”) type

After Removing anyref

ref.is_null <*reftype*> : [<*reftype*>] → []

type annotation to make type unique

Background

Original Wasm design goal:
every instruction's type is self-contained

That is, every operand type is

- either apparent from instruction + immediates,
- or fully polymorphic (e.g., drop)

The Future

There will be many instructions of this sort

ref.is_null
ref.as_non_null
br_on_null
call_ref
func.bind
struct.new_with_rtt
struct.get
struct.set
array.new_with_rtt
array.get
array.set
array.len
rtt.sub
ref.test
ref.cast
br_on_cast
...

The Future

There will be many instructions of this sort

Type annotations increase code size

Redundant annotations increase type checking time (need to compare more types)

Though we don’t know if that matters in practice

Reference Types Proposal

With anyref, we could have deferred this discussion

Without, it becomes relevant to this proposal

Could leave ref.is_null as is,
but it will likely become an outlier

I propose to remove the type annotation

Proposal

ref.is_null : [<*reftype*>] → []

instruction is polymorphic, but only over refs

linear validator just looks it up on stack,
like for drop, and verifies it's a reference

very small change in implementations

Poll

Remove type annotation from ref.is_null?