GC & Typed Function References

Proposal update

Andreas Rossberg

Recap: Function references

Add new value type: (ref null? $t)

Subtyping: (ref $t) <: (ref null $t)

Instructions:

call_ref, br_on_(non_)null, ref.as_non_null

Non-null refs do not have a default value

…validation tracks initialisation status of locals

…tables require an initialisation value

Recap: GC (1)

Add new type defs: (**struct** (field t*)), (**array** t*)

Explicit subtyping: (**type** $t (**sub** $s (struct ...)))

...also allowed for function types

Hierarchy of reference types: struct, array, i31 <: eq <: any

Mutually recursive types:

```
(rec
  (type $a (struct (field i32 (ref $b) i32)))
  (type $b (array (ref null $a)))
)
```

Recap: GC (2)

Reference instruction: ref.eq

Struct instructions: struct.new/get/set …

Array instructions: array.new/get/set/len …

Tagged integers: ref.i31, i31.get

Cast instructions as escape hatch:
ref.test, ref.cast, br_on_cast, br_on_cast_fail

Conversion instructions for externref:
extern.internalize/externalize

Status

Both proposals were voted to phase 3 at last F2F meeting (2022/10/26)

Proposals have been mostly stable since

Implemented in V8 and SpiderMonkey

Changes since Phase 3 vote

Added return_call_ref (consequence of moving tail calls to phase 4)

Added bulk array instructions: array.fill/copy, array.init_data/elem

Added final attribute for subtype declarations

…Wasm 1.0 type definitions reinterpreted as final

Added 2nd type annotation on br_on_cast/_fail and tweaked typing

Finalised opcode assignments

Tail calls to references

return_call_ref $\$t$: t_1* (ref null $\$t$) → ⊥

where $\$t$ = func t_1* → t_2*

∧ current function : t_3* → t_2*

Bulk array instructions

array.fill $\$t$: (ref null $\$t$) i32 t_{elem} i32 → []

array.copy $\$t_1$ $\$t_2$: (ref null $\$t_1$) i32 (ref null $\$t_2$) i32 i32 → []

array.init_data $\$t$ $\$d$: (ref null $\$t$) i32 i32 i32 → []

array.init_elem $\$t$ $\$e$: (ref null $\$t$) i32 i32 i32 → []

Final types

(**type** $t1 (**sub open** (**struct** (field i32 i32))))

(**type** $t2 (**sub open** $t1 (**struct** (field i32 i32 i32))))

(**type** $t3 (**sub final** $t1 (**struct** (field i32 i32 i32))))

(**type** $t (**struct** (field i32)))
= (**type** $t (**sub final** (**struct** (field i32))))

Indirect calls

(**type** $t (**func** …)) ;; = (**type** $t (**sub final** (**func** …)))

call_indirect (type $t) ;; as fast as before (type equality check)

(**type** $s (**sub open** (**func** …)))

call_indirect (type $s) ;; may be slower (subtype check)

Casts

br_on_cast $*l* t_{from} t_{to} : $t_{from} \rightarrow t'_{from}$ where $l : t_{to}$

br_on_cast_fail $*l* t_{from} t_{to} : $t_{from} \rightarrow t_{to}$ where $l : t'_{from}$

where $t'_{from} = t_{from} \setminus t_{to}$ (t_{from} without null cases included in t_{to})

br_on_cast $*l* (ref null $t) (ref null $t') : (ref null $t) → (ref $t)

Ensures a principal types property that implies that no context knowledge is needed to *validate* or *execute* a given instruction

Spec

Spec authoring tracking issue #376

Open · 45 of 53 tasks · rossberg opened this issue on May 14 · 0 comments

rossberg commented on May 14 • edited · Member

- [ ] Syntax
  - [x] Heap types (Spec abstract syntax for GC types and instructions #373)
  - [x] Type definitions (Validation of instructions (plus some infra and fixes) #377)
  - [x] Instructions (Spec abstract syntax for GC types and instructions #373)
  - [ ] Change syntax of `final` attribute (Use `open` instead of `final` in the text format #413)
- [x] Validation
  - [x] Heap types (Spec type and module validation, subtyping, and module instantiation. #382)
  - [x] Type definitions (Spec type and module validation, subtyping, and module instantiation. #382)
  - [x] Instructions (Validation of instructions (plus some infra and fixes) #377)
  - [x] Type equivalence (Spec type and module validation, subtyping, and module instantiation. #382)
  - [x] Subtyping (Spec type and module validation, subtyping, and module instantiation. #382)
  - [x] Relax global initialisation ordering (Spec sequential visibility for globals #420)
  - [ ] ~~Relax br_on_cast (blocked on Can the rt2 <: rt1 constraint on br_on_cast* instructions be relaxed? #381)~~
- [ ] Execution
  - [x] Instructions, formal (Spec execution, part 1 #374)
  - [x] Instructions, prose (Prose for basic struct/array instructions #393)
  - [x] Instantiation (Spec type and module validation, subtyping, and module instantiation. #382)
  - [ ] Perform type unrolling (Extend soundness appendix #406)
  - [x] Relax global initialisation ordering (Spec sequential visibility for globals #420)
- [x] Binary format
  - [x] Heap types (Spec binary format #383)
  - [x] Type definitions (Spec binary format #383)
  - [x] Instructions (Spec binary format #383)
  - [x] Opcode reordering (Final opcodes #372)
- [x] Text format
  - [x] Heap types (Spec text format #385)
  - [x] Type definitions (Spec text format #385)
  - [x] Instructions (Spec text format #385)
  - [x] Dependent field indices (Spec text format #385)
- [ ] Appendices
  - [x] Embedder interface
  - [x] Algorithm (Extend validation algorithm appendix #390)
  - [ ] Properties (Extend soundness appendix #406)
  - [ ] Store wf should exclude impossible cycles (cf. Hosts should be disallowed from creating impossible data #332)
  - [x] Extended name section (Extends Name Section w/ Type & Field #415)
  - [x] Changes (Add Changes section for GC #388)
- [x] Indices
  - [x] Type opcodes (Extend type/instr/rule indices #389)
  - [x] Instruction opcodes (Extend type/instr/rule indices #389)
  - [x] Rules (Extend type/instr/rule indices #389)
  - [x] Opcode reordering (Final opcodes #372)
- [x] JS API
  - [x] Tweak handling of externtype conversions (Update JS API #379)
- [x] Bulk array instructions
  - [x] Syntax (Spec syntax, text, binary, and execution for bulk array ops #387)
  - [x] Validation (Bulk array validation #401)
  - [x] Execution (Spec syntax, text, binary, and execution for bulk array ops #387)
  - [x] Execution, prose (Execution prose for bulk array operations #408)
  - [x] Binary format (Spec syntax, text, binary, and execution for bulk array ops #387)
  - [x] Text format (Spec syntax, text, binary, and execution for bulk array ops #387)
  - [x] Changes (Add Changes section for GC #388)
  - [x] Indices (Extend type/instr/rule indices #389)
  - [x] Todo about handling packed fields correctly with array.copy (Update array.copy semantics to handle packed types #416)

6

Status

✅ Specification

✅ Reference interpreter

✅ Test suite (open PR for JS tests)

✅ Implemented in V8 & SpiderMonkey

Phase 3 (2022/10/26)

No open issues

…open PR for changing `final` syntax

https://github.com/WebAssembly/gc/issues/371

Poll

Move to phase 4?