Stack Inspection

Ross Tate

Stack Traces

- Java: Thread.currentThread().getStackTrace() : StackTraceElement[]
- C#: new StackTrace(true).GetFrames() : StackFrame[]
- Python: traceback.extract_stack() : (String,int,String,String)[]

- Stack traces are in terms of *source* code, not wasm code

Debugging

- https://www.jetbrains.com/help/idea/examining-suspended-program.html

Linear-Memory Garbage Collection

- stack inspection is used to collect (linear memory) roots
- stack inspection is used to update moved (linear memory) roots

Basic Instructions

call_stack $dispatch_tag : [ti*] -> [to*]

- $dispatch_tag : [ti*] -> [to*]
- looks up the stack
  for an "answer" for $dispatch_tag

answer $dispatch_tag instr1* within instr2* end

- $dispatch_tag : [ti*] -> [to*]
- instr1* : [ti*] -> [to*]
- executes instr2*, during which
  answering any call_stack $dispatch_tag
  with instr1*

Two-Phase Exception Handling

C#

```
bool flag = …;
try {
   …
} catch (Exception) when (flag = !flag) {
   println(“caught first”);
} catch (Exception) when (flag = !flag) {
   println(“caught second”);
}
```

WebAssembly

```
(block $outer
   (block $first
      (block $second
         (answer $csharp_exn ;; [csharp_ref] -> []
            (local.set $flag (i32.xor 1 (local.get $flag)))
            (unwinding br_if $first (local.set $flag))
            (local.set $flag (i32.xor 1 (local.get $flag)))
            (unwinding br_if $second (local.set $flag))
            (call_stack_from $outer $csharp_exn)
         within
            …
            br $outer
         )
      ) ;; $second : [csharp_ref]
      (call $println “caught second”)
      (br $outer)
   ) ;; $first : [csharp_ref]
   (call $println “caught first”)
   (br $outer)
) ;; $outer : []
```

Stack Walking

- java.lang.StackWalker.walk<T>(Stream<StackFrame> → T) : T