Exception Handling: New Instruction Format

Heejin Ahn

Recap: Motivation for Reintroducing `exnref`

- We encountered issues encountered after Phase 3
  - Difficulty in handling the identity of thrown exceptions in the JS API spec.
  - Complexity in the formal specification and engines.
  - Inability to CPS-transform (e.g. asyncify) Wasm functions with exceptions.
  - Lack of **`exnref`** significantly restricts toolchains' code transformations and sometimes leads to unnecessary duplication. (Specific feedback from J2CL authors).
- Proximate cause: new form of storage in the form of lexical rethrow
  - Unlike other existing forms of storage in Wasm

Proposed Instruction Format

```
try blocktype (catch $i $li) (catch $j $lj) … (catch_all $l)
  instruction*
end
```

- No `try`-`catch`-...-`catch_all` blocks anymore
  - More in line with Wasm regular control flow
- `catch` labels are bundled with `try` → simpler decoding
- No need for `catch` and `catch_all` opcodes
  - Handler list can be a vector of immediates as in `br_table`

Proposed Instruction Format: Handler Example

```
block $li (result …, exnref)
  block $lj  (result …, exnref)
    block $l (result exnref)
      try blocktype (catch i $li) (catch j $lj) (catch_all $l)
        instruction*
      end
    end
    handler code for catch_all
  end
  handler code for catch j
end
handler code for catch i
```

Changes for Other Instructions

- `rethrow` will take an explicit `exnref`, not an immediate
  - Not dependent on lexical surroundings anymore, which was a pain
- `try-delegate` will be removed
  - `exnref` can achieve same semantics

Migration Plan

- VMs will support both current + new spec for the foreseeable future
- We will provide a binary translator to facilitate migration
- Emscripten (+ LLVM / Binaryen) toolchain will support emitting new binary under a flag first, and then switch when the implementation is stabilized
    - Toolchain will support emitting new binary sooner using the translator, even before all implementation is done
- Deprecation (removal) of old instructions will only happen the usage is low enough

Github Discussions

- Reintroducing exnref
  - https://github.com/WebAssembly/exception-handling/issues/280
- Proposed instruction format for reintroducing exnref
  - https://github.com/WebAssembly/exception-handling/issues/281