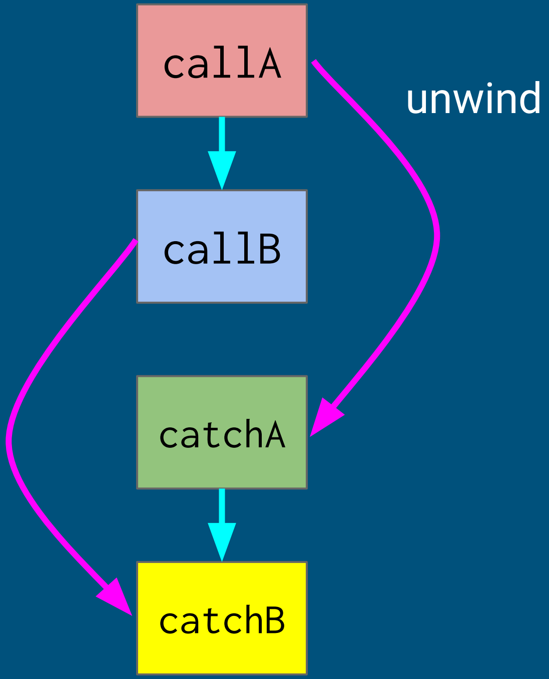
# Extensibility for Two-phase Unwinding in Exception Handling

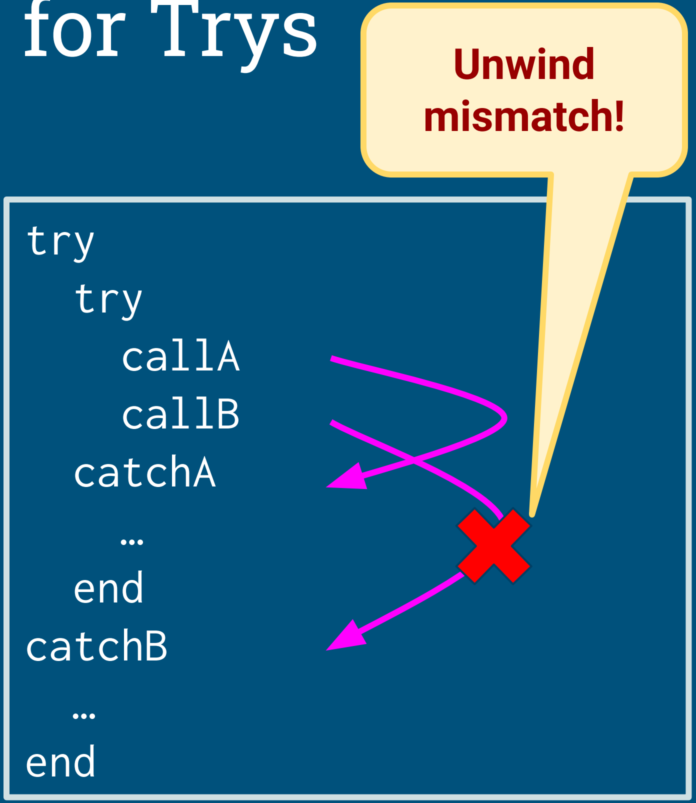
Heejin Ahn

# Recap for **`exnref`**

- `exnref` was introduced in 2018 for two reasons:
  - Fix unwind mismatch problem in toolchain
  - To be extend to `evtref` in future resumable EH proposal
- But both reasons are not valid anymore
  - To support two-phase unwinding, we cannot use `exnref` to fix unwind mismatch problem
  - Typed continuation proposal does not use `evtref` anymore

# Review: CFG Stackification for Trys

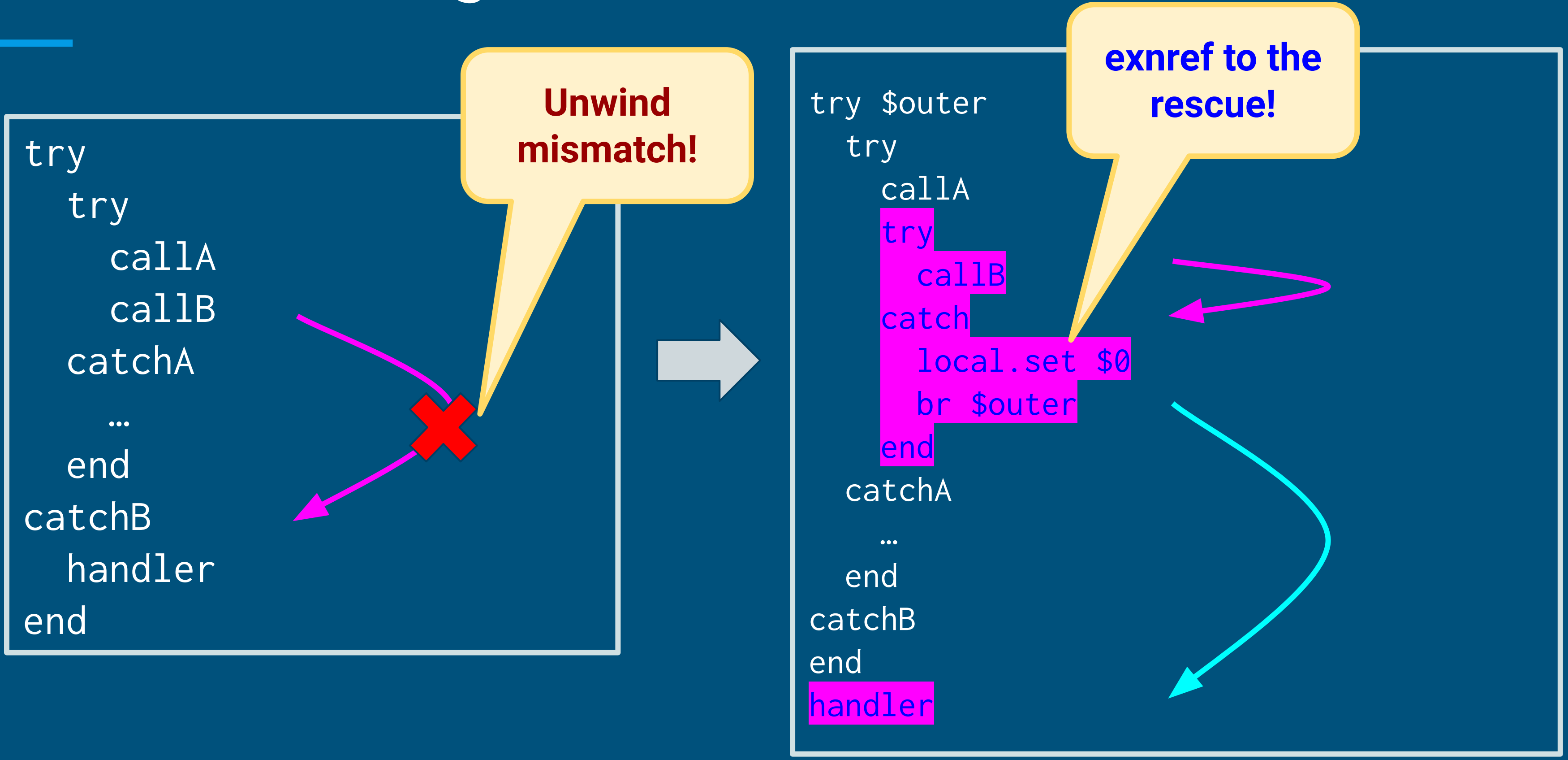




Unwind mismatch!

```
try
  try
    callA
    callB
  catchA
    …
  end
catchB
  …
end
```

# Review: Fixing Unwind Mismatches



```
try
  try
    callA
    callB
  catchA
    …
  end
catchB
  handler
end
```

Unwind mismatch!

```
try $outer
  try
    callA
    try
      callB
    catch
      local.set $0
      br $outer
    end
  catchA
    …
  end
catchB
end
handler
```

exnref to the rescue!

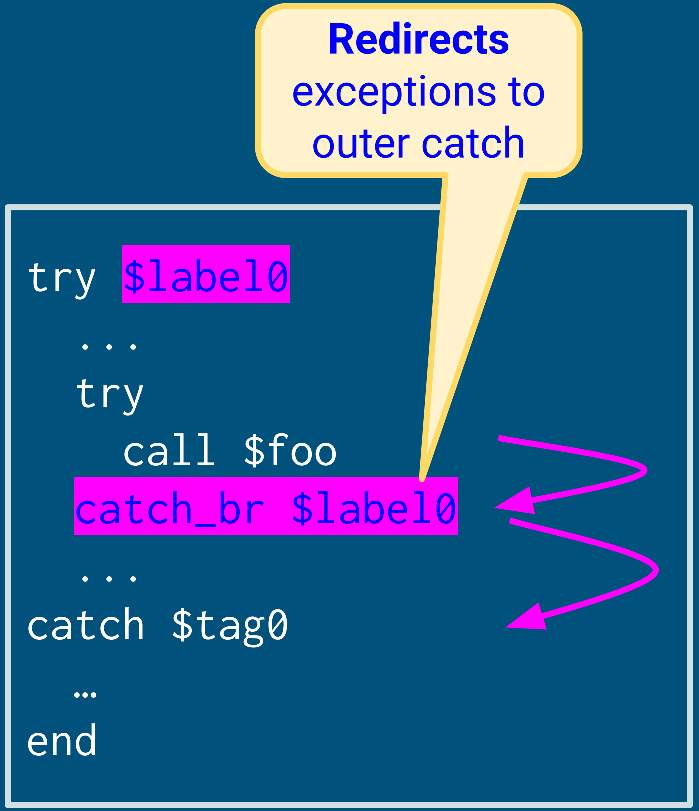
# Suggested Changes

- Remove `exnref` and go back to the first proposal, with some additions:
  - `catch_br` instruction
  - Splitting `catch` and `unwind`

```
try
  …
catch $tag0
  …
catch $tag1
  …
catch_all
  …
end
```

# Add catch_br Instruction

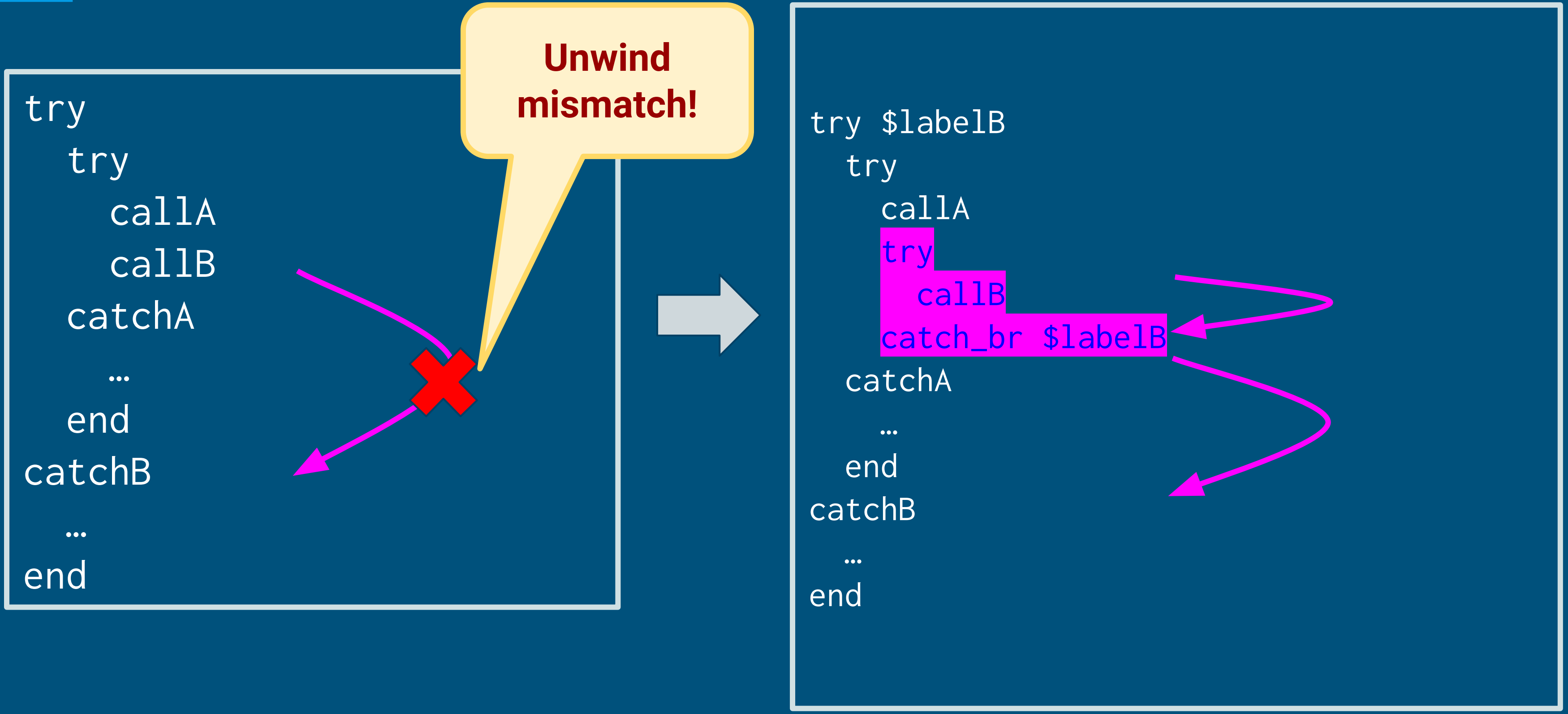
- Does not have a tag or a filter
- Only *redirects* exceptions to another catch in outer scope
- Does not have a catch body and thus no matching end instruction
- Can be examined *statically* by first phase search



**Redirects** exceptions to outer catch

```
try $label0
  ...
  try
    call $foo
  catch_br $label0
  ...
catch $tag0
  …
end
```

# Fixing Unwind Mismatches Revisited



```
try
  try
    callA
    callB
  catchA
    …
  end
catchB
  …
end
```

Unwind mismatch!

```
try $labelB
  try
    callA
    try
      callB
    catch_br $labelB
  catchA
    …
  end
catchB
  …
end
```

# Split `catch` and `unwind`

- Second (cleanup) phase runs destructors while unwinding the stack
- We need a way to *resume* the second phase after running destructors
  - `rethrow` is different; it starts a new two phase search
- At the end of unwind block, the second phase automatically resumes
- Unlike catch, `unwind` does not take a tag or a filter (in the presence of two-phase)

```
try
  …
unwind
  destructor code
end
```

# Restore rethrow's Immediate Arg?

- In first proposal, `rethrow` used have an imm arg specifying which exception to rethrow
- Do we restore this behavior or not?
    - If not, rethrow will not have any arg and rethrow the most recent exception

```
try
  ...
catchA
  try
    ...
  catchB
    ...
    rethrow N
  end
end
```

**Which to rethrow? catchA or catchB?**

# Discussions

# Poll: Should we adopt the proposed change?