Annotations

Status Update and Poll

Andreas Rossberg

Motivation

Custom annotations as textual equivalent to custom sections

bring text format on par with binary format

human-readable and -writable

Allow round-tripping between binary and text format

robust wrt encoding variations and manual edits

Use cases e.g. branch hinting proposal

Proposal Summary

Free-form format, just like custom sections

(@id ...) anywhere in text

where id corresponds to custom section id

No prescribed syntax or semantics, contents is arbitrary token sequence

Can be ignored or recognised by engines and tools

Only affects text format

New: incorporated quoted symbolic identifiers into this proposal

\$"module" \$"spaces spaces\tspaces" \$"λ"

Appendix

Custom section appendix defines name annotations as counterpart for name section

```
(func (@name "λ") (param i32) ...)
```

Also defines generic representation of custom sections

```
(module (@custom "id" (after func) "bytes") ...)
```

...usable as fallback, but may fail to round-trip correctly (unfortunately, some custom section formats depend on binary encoding details)

Status

Phase 3 (since 2020/03)

- ✓ Spec
- Reference interpreter with new plug-in infrastructure for custom sections/annotations (with help from Yuri)
- ▼ Test suite
 with new infrastructure for custom tests and selective opt-in
- ✓ Implementation in Binaryen (except @name annotations) and wasm-tools (with @name)

Dependency for branch hinting

Poll

Advance to phase 4?