Annotations for the Wasm text format

Proposal

Andreas Rossberg

Dfinity

Motivation

Binary format has custom sections

No equivalent in text format

…cannot round-trip custom sections

…cannot express meta information
(i.e., for host binding, debugging, etc.)

Proposal Summary

Allow (@id …) annotations anywhere in text

No prescribed syntax or semantics

Can be ignored by tools

…but some may assign meaning to some ids

Only affects text format

Syntax

Allow annotation brackets anywhere in text format

```
annot ::= "(@"id  annotelem* ")"
```

Headed by id that categorises extension

…by convention, corresponds to name of a custom section

Body is arbitrary token sequence, but must be well-bracketed

```
annotelem ::=
    keyword | reserved | uN | sN | fN | string | "$"id |
    "(" annotelem* ")" |
    "(@"id annotelem* ")"
```

A tool will define more concrete structure for its annotations

Example:
Generic custom section

```
(module
  ...
  (@custom "my-section" (after function)
    "contents-bytes"
  )
)
```

Example: JS host bindings

```
(func (export "f") (param i32 (@js unsigned)) ...)

(func (export "method")
  (param $x anyref (@js this)) (param $y i32) ...
)

(func (import "m" "constructor")
  (@js new) (param i32) (result anyref) ...
)
```