e ::= (unreachable) | (nop) | (drop) | (select)
| (block tf (e ...)) | (loop tf (e ...))
| (if tf (e ...) else (e ...)) | (br i) | (br-if i)
| (br-table (i ...)) | (return) | (call i)
| (call-indirect tf) | (get-local i) | (set-local i)
| (tee-local i) | (get-global i) | (set-global i)
| (t load a o) | (t load (tp sx) a o) | (t store a o)
| (t store (tp) a o) | (current-memory) | (grow-memory)

| (inn $iunop$) | (fnn $funop$)
| (inn $ibinop$) | (fnn $fbinop$)
| (inn $itestop$)
| (inn $irelop$) | (fnn $frelop$)
| (t $cvtop$ t) | (t $cvtop$ t sx)

| (i32 const (side-condition $integer_1$ (u32? (term $integer_1$))))
| (i64 const (side-condition $integer_1$ (u64? (term $integer_1$))))
| (f32 const (side-condition $real_1$ (single-flonum? (term $real_1$))))
| (f64 const (side-condition $real_1$ (double-flonum? (term $real_1$))))

inn ::= i32 | i64
fnn ::= f32 | f64
t ::= i32 | i64 | f32 | f64
tp ::= i8 | i16 | i32
tf ::= ((t ...) -> (t ...))
mut ::= const | var
tg ::= (mut t)
sx ::= signed | unsigned
$unop$ ::= $iunop$ | $funop$
$binop$ ::= $ibinop$ | $fbinop$
$testop$ ::= $itestop$
$relop$ ::= $irelop$ | $frelop$
$iunop$ ::= clz | ctz | popcnt
$ibinop$ ::= add | sub | mul | div-s | div-u | rem-s | rem-u
| and | or | xor | shl | shr-s | shr-u | rotl | rotr
$itestop$ ::= eqz
$irelop$ ::= eq | ne | lt-s | lt-u | gt-s | gt-u | le-s | le-u | ge-s | ge-u
$funop$ ::= abs | neg | sqrt | ceil | floor | nearest
$fbinop$ ::= add | sub | mul | div | min | max | copysign
$frelop$ ::= eq | ne | lt | gt | le | ge
$cvtop$ ::= convert | reinterpret
i, j, n, m ::= $natural$
a, o ::= (side-condition $natural_1$ (u32? (term $natural_1$)))
c ::= $real$
f ::= (func (ex ...) tf (local (t ...) (e ...)))
| (func (ex ...) tf im)
$glob$ ::= (global (ex ...) tg (e ...))
| (global (ex ...) tg im)
tab ::= (table (ex ...) n (i ...))
| (table (ex ...) n im)
mem ::= (memory (ex ...) n)
| (memory (ex ...) n im)
im ::= (import $string$ $string$)
ex ::= (export $string$)
mod ::= (module (f ...) ($glob$ ...) (tab ...) (mem ...))