

Grammar

Version 6.12.0.4

February 28, 2018

```
program = toplevel-binding
        ...

toplevel-binding = (define-signature name = sig-expr)
                  | (define-module name = mod-expr)

sig-expr = name
          | (sig component-decl ...)
          | (Π ([name : sig-expr]) sig-expr)
          | (let ([name mod-expr]) sig-expr)

mod-expr = name
          | (mod component-def ...)
          | (seal mod-expr :> sig-expr)
          | (mod-expr mod-expr)
          | (λ ({name : sig-expr}) mod-expr)
          | (let ([name mod-expr]) mod-expr)

component-decl = (type name)
                | (type name = T)
                | (val name : T)

component-def = (type name = T)
                | (val name = E)
                | (val name : T = E)

T = T*
  | (∀ (name ...) T*)

T* = Int
   | (-> T* ... T*)
```

```
E = integer  
  | (E E)  
  | (λ ({name : T*} ...) E)  
  | (Λ (name ...) E)
```