Expr
$e ::= c \mid e_1 \oplus e_2$

Const $c ::= n \mid s \mid b$

Binop $\oplus ::= + \mid * \mid \hat{}$

Integer n
String s
Bool $b ::= true \mid false$

$1 + 2$
"hello"

$\frac{}{c \text{ val}}$ (parse n)

Val is a relation on Expr

$\forall c \in Const, c \text{ val}.$

$[\![n]\!] = n$

$[\![e_1 + e_2]\!] = [\![e_1]\!] + [\![e_2]\!]$

$\frac{e_1 \Downarrow n_1 \quad e_2 \Downarrow n_2 \quad n_3 = n_1 + n_2}{e_1 + e_2 \Downarrow n_3}$

```
BigStep(
  Add(e1,e2),
  n3 ) :- ...
```

$\frac{e_1 \Downarrow \text{error}}{e_1 + e_2 \Downarrow \text{error}}$

$\frac{e_1 \hookrightarrow e_1' \quad e_2 \text{ val}}{e_1 \oplus e_2 \hookrightarrow e_1' \oplus e_2}$ E-Add-1

$(1+2)+(3+4)$

$(1+2)+3 \hookrightarrow 3+3$

$\frac{e_1 \text{ val} \quad e_2 \hookrightarrow e_2'}{e_1 \oplus e_2 \hookrightarrow e_1 \oplus e_2'}$ E-Add-2

$\frac{n_3 = n_1 + n_2}{n_1 + n_2 \hookrightarrow n_3}$ E-Add-3

Type $\tau ::=$
int | bool | string

$$\frac{s_3 = s_1 \hat{} s_2}{s_1 \hat{} s_2 \hookrightarrow s_3}$$

$$\frac{}{n : \text{int}}\ \text{T-Num} \qquad \frac{}{s : \text{string}}$$

$$\frac{}{b : \text{bool}}$$

$$\frac{\frac{}{1 : \text{int}} \quad \frac{}{2 : \text{int}}}{1+2 : \text{int}}\ \text{T-Add} \qquad \frac{}{3 : \text{int}}\ \text{T-Num}$$

$$\frac{(1+2)+3 : \text{int}}{}\ \text{T-Add}$$

$$\frac{e_1 : \text{int} \quad e_2 : \text{int}}{e_1 + e_2 : \text{int}}\ \text{T-Add}$$

Expr $e ::=$
...
$| (e_1, \ldots e_n)$
$| e.i$

Nat $i \in \mathbb{N}$

$$\frac{e_1 \hookrightarrow e_1'}{(e_1, \ldots e_n) \hookrightarrow (e_1', \ldots e_n)}$$

Type $\tau ::=$
...
$| (\tau_1, \ldots, \tau_n)$

Expr $e ::= (e_1, e_2)$
$\tau ::= (\tau_1, \tau_2)$

$$\frac{\forall_{i \in [n]}\ e_i : \tau}{(e_1, \ldots e_n) : (\tau, \ldots \tau)}$$

$$\frac{\forall_{i \in [n]}\ e_i\ \text{val}}{(e_1, \ldots e_n)\ \text{val}}$$

$$\frac{\forall_{i \in [n]}\ e_i : \tau_i}{(e_1, \ldots e_n) : (\tau_1, \ldots \tau_n)}$$

$$\frac{e : (\tau_1, \ldots \tau_n) \quad i \in [n]}{e.i : \tau_1}$$

$(3, \text{"hi"}).2 + 1 \hookrightarrow \text{"hi"} + 1$

Type safety: a well-typed expression can't go wrong

Progress:

if $e : \tau$, then either e val or $e \hookrightarrow e'$

Preservation:

if $e : \tau$ AND $e \hookrightarrow e'$ then $e' : \tau$

$e =$ 1 as float : float

$\hookrightarrow$ 1.0 : float

Expr (e) ::= ... | let x = e_1 in e_2 | x

Variable (x)

let x = 1 in x+1

$\overset{*}{\hookrightarrow}$ 2

e_1 val

---

let x = e_1 in $e_2 \hookrightarrow e_2[x \mapsto e_1]$

$(x+1)[x \mapsto 2] = 2+1$

$(\text{let } x = 1 \text{ in } x)[x \mapsto 2] = \text{let } x = 1 \text{ in } x$

$$\frac{\Gamma \vdash e_1 : \tau_1 \quad \Gamma, x : \tau_1 \vdash e_2 : \tau_2}{\Gamma \vdash \text{let } x = e_1 \text{ in } e_2 : \tau_2}$$

$$\frac{\Gamma[x] = \tau}{\Gamma \vdash x : \tau}$$

Ty Ctxt $\Gamma ::=$
$\{ \text{Var} \mapsto \text{Type} \}$