

Quiz

Question #1: What is the value of the following expression?

{ + 1 2 }

Wrong answer: **0**

Wrong answer: **42**

Answer: **3**

Quiz

Question #2: What is the value of the following expression?

```
{+ fun 17 8}
```

Wrong answer: error

Answer: Trick question! `{+ fun 17 8}` is not an expression

Language Grammar for Quiz

```
<MFAE> ::= <num>
          | true
          | false
          | {+ <MFAE> <MFAE>}
          | {- <MFAE> <MFAE>}
          | {= <MFAE> <MFAE>}
          | <id>
          | {fun {<id>*} <MFAE>}
          | {<MFAE> <MFAE>*}
          | {if <MFAE> <MFAE> <MFAE>}
```

Quiz

Question #3: Is the following an expression?

```
{{fun {} 1} 7}
```

Wrong answer: **No**

Answer: **Yes** (according to our grammar)

Quiz

Question #4: What is the value of the following expression?

```
{{fun {} 1} 7}
```

Answer: `1` (according to some interpreters)

But no *real* language would accept `{{fun {} 1} 7}`

Let's agree to call `{{fun {} 1} 7}` an ***ill-formed expression*** because `{fun {} 1}` should be used with only zero arguments

Let's agree to never evaluate ill-formed expressions

Quiz

Question #5: What is the value of the following expression?

```
{{fun {} 1} 7}
```

Answer: None — the expression is ill-formed

Quiz

Question #6: Is the following a well-formed expression?

```
{+ {fun {} 1} 8}
```

Answer: Yes

Quiz

Question #7: What is the value of the following expression?

```
{+ {fun {} 1} 8}
```

Answer: None — it produces an error:

+: expects a numV, given a closureV

Let's agree that a **fun** expression cannot be inside a **+** form

Quiz

Question #8: Is the following a well-formed expression?

```
{+ {fun {} 1} 8}
```

Answer: **No**

Quiz

Question #9: Is the following a well-formed expression?

```
{+ {{fun {x} x} 7} 5}
```

Answer: Depends on what we meant by *inside* in our most recent agreement

- *Anywhere inside* — **No**
- *Immediately inside* — **Yes**

Since our interpreter produces **12**, and since that result makes sense, let's agree on *immediately inside*

Quiz

Question #10: Is the following a well-formed expression?

```
{+ {{fun {x} x} {fun {y} y}} 5}
```

Answer: **Yes**, but we don't want it to be!

Quiz

Question #11: Is it possible to define **well-formed** (as a decidable property) so that we reject all expressions that produce errors?

Answer: Yes: reject *all* expressions!

Quiz

Question #12: Is it possible to define **well-formed** (as a decidable property) so that we reject *only* expressions that produce errors?

Answer: No

```
{+ 1 {if ... 1 {fun {x} x}}}
```

If we always knew whether . . . produces true or false, we could solve the halting problem

Types

Solution to our dilemma:

In the process of rejecting expressions that are certainly bad, also reject some expressions that are good

```
{+ 1 {if {prime? 131101}
         1
         {fun {x} x}}}}
```

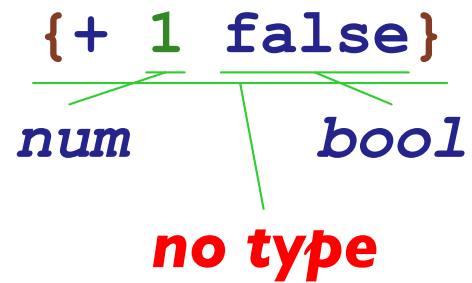
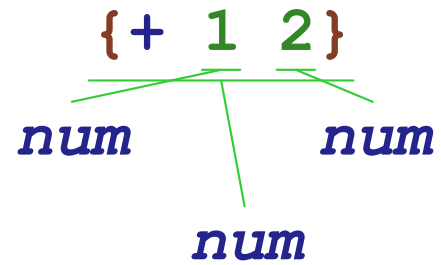
Overall strategy:

- Assign a **type** to each expression *without evaluating*
- Compute the type of a complex expression based on the types of its subexpressions

Types

`1` : `num`

`true` : `bool`



Type Rules

$\langle \text{num} \rangle : \text{num}$

$\text{true} : \text{bool}$

$\text{false} : \text{bool}$

$\langle \text{MFAE} \rangle_1 : \text{num}$

$\langle \text{MFAE} \rangle_2 : \text{num}$

$\{ + \langle \text{MFAE} \rangle_1 \langle \text{MFAE} \rangle_2 \} : \text{num}$

$1 : \text{num}$

$\text{true} : \text{bool}$

$1 : \text{num}$

$2 : \text{num}$

$\{ + 1 2 \} : \text{num}$

$1 : \text{num}$

$\text{false} : \text{bool}$

$\{ + 1 \text{false} \} : \text{no type}$

Type Rules

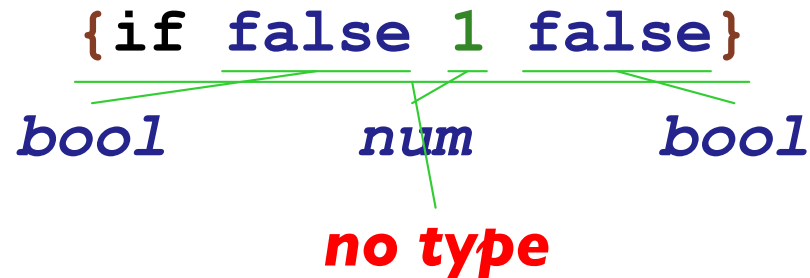
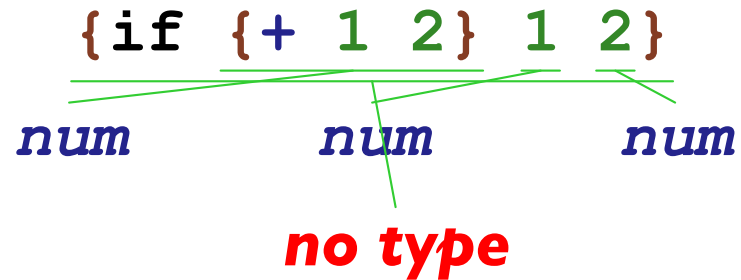
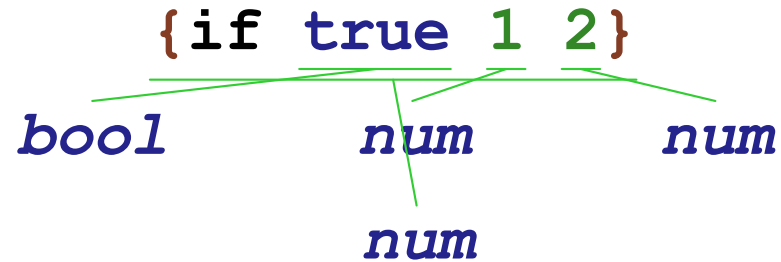
$\langle \text{num} \rangle : \text{num}$

$\text{true} : \text{bool}$

$\text{false} : \text{bool}$

$$\frac{\langle \text{MFAE} \rangle_1 : \text{num} \quad \langle \text{MFAE} \rangle_2 : \text{num}}{\{ + \langle \text{MFAE} \rangle_1 \langle \text{MFAE} \rangle_2 \} : \text{num}}$$
$$\frac{1 : \text{num} \quad 2 : \text{num}}{\{ + 1 2 \} : \text{num}} \quad 3 : \text{num}$$
$$\frac{\{ + 1 2 \} : \text{num} \quad 3 : \text{num}}{\{ + \{ + 1 2 \} 3 \} : \text{num}}$$

Types: Conditionals



Conditional Type Rules

$\langle \text{MFAE} \rangle_1 : \text{bool}$ $\langle \text{MFAE} \rangle_2 : \langle \text{type} \rangle_0$ $\langle \text{MFAE} \rangle_3 : \langle \text{type} \rangle_0$

$\{\text{if } \langle \text{MFAE} \rangle_1 \langle \text{MFAE} \rangle_2 \langle \text{MFAE} \rangle_3\} : \langle \text{type} \rangle_0$

$\text{true} : \text{bool}$ $1 : \text{num}$ $2 : \text{num}$

$\{\text{if true } 1 \ 2\} : \text{num}$

$\{+ \ 1 \ 2\} : \text{num}$ $1 : \text{num}$ $2 : \text{num}$

$\{\text{if } \{+ \ 1 \ 2\} \ 1 \ 2\} : \text{no type}$

$\text{false} : \text{bool}$ $1 : \text{num}$ $\text{false} : \text{bool}$

$\{\text{if false } 1 \ \text{false}\} : \text{no type}$

Types: Variables and Functions

x : no type

`{fun {x : bool} x}`

bool

(bool → bool)

`{fun {x : bool} {if x 1 2}}`

bool

num

num

num

(bool → num)

Variable and Function Type Rules

$$[\dots \langle \text{id} \rangle \leftarrow \tau \dots] \vdash \langle \text{id} \rangle : \tau$$
$$\Gamma [\langle \text{id} \rangle \leftarrow \tau_1] \vdash \mathbf{e} : \tau_2$$

$$\Gamma \vdash \{ \text{fun } \{ \langle \text{id} \rangle : \tau_1 \} \mathbf{e} \} : (\tau_1 \rightarrow \tau_2)$$

Abbreviations: $\tau = \langle \text{type} \rangle$
 $\mathbf{e} = \langle \text{MFAE} \rangle$
 $\Gamma = \langle \text{env} \rangle$

Variable and Function Type Rules

$$[\dots \langle \text{id} \rangle \leftarrow \tau \dots] \vdash \langle \text{id} \rangle : \tau$$
$$\Gamma [\langle \text{id} \rangle \leftarrow \tau_1] \vdash \mathbf{e} : \tau_2$$

$$\Gamma \vdash \{ \text{fun } \{ \langle \text{id} \rangle : \tau_1 \} \mathbf{e} \} : (\tau_1 \rightarrow \tau_2)$$
$$\emptyset \vdash \mathbf{x} : \text{no type}$$
$$[\mathbf{x} \leftarrow \text{bool}] \vdash \mathbf{x} : \text{bool}$$

$$\emptyset \vdash \{ \text{fun } \{ \mathbf{x} : \text{bool} \} \mathbf{x} \} : (\text{bool} \rightarrow \text{bool})$$
$$[\mathbf{x} \leftarrow \text{bool}] \vdash \mathbf{x} : \text{bool} \quad [\mathbf{x} \leftarrow \text{bool}] \vdash \mathbf{1} : \text{num} \quad [\mathbf{x} \leftarrow \text{bool}] \vdash \mathbf{2} : \text{num}$$

$$[\mathbf{x} \leftarrow \text{bool}] \vdash \{ \text{if } \mathbf{x} \ \mathbf{1} \ \mathbf{2} \} : \text{num}$$

$$\emptyset \vdash \{ \text{fun } \{ \mathbf{x} : \text{bool} \} \{ \text{if } \mathbf{x} \ \mathbf{1} \ \mathbf{2} \} \} : (\text{bool} \rightarrow \text{num})$$

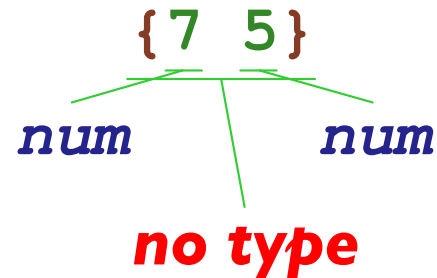
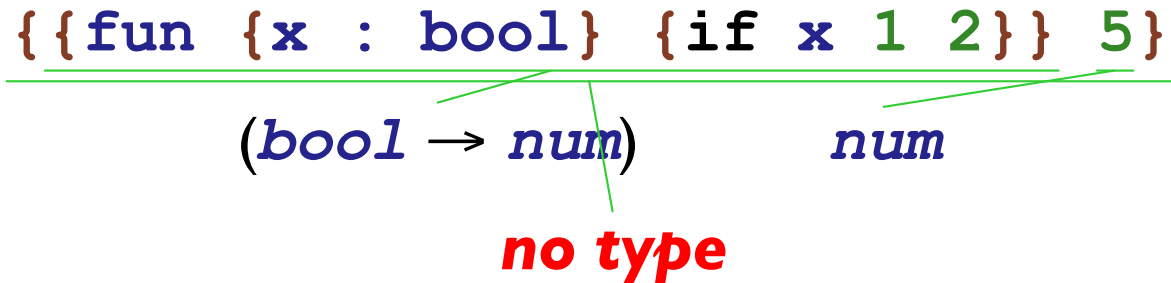
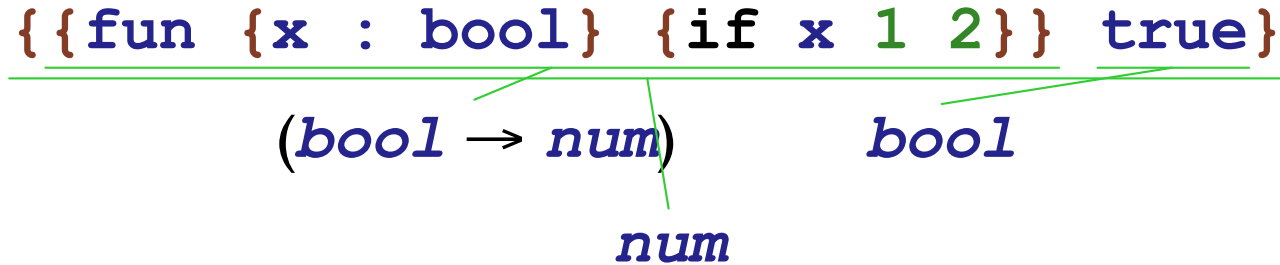
Revised Rules

$$\Gamma \vdash \langle \text{num} \rangle : \text{num}$$
$$\Gamma \vdash \text{true} : \text{bool}$$
$$\Gamma \vdash \text{false} : \text{bool}$$
$$\Gamma \vdash \mathbf{e}_1 : \text{num} \quad \Gamma \vdash \mathbf{e}_2 : \text{num}$$

$$\Gamma \vdash \{+ \mathbf{e}_1 \ \mathbf{e}_2\} : \text{num}$$
$$\Gamma \vdash \mathbf{e}_1 : \text{bool} \quad \Gamma \vdash \mathbf{e}_2 : \tau_0 \quad \Gamma \vdash \mathbf{e}_3 : \tau_0$$

$$\Gamma \vdash \{\text{if } \mathbf{e}_1 \ \mathbf{e}_2 \ \mathbf{e}_3\} : \tau_0$$

Types: Function Calls



Function Call Type Rule

$$\Gamma \vdash \mathbf{e}_1 : (\tau_2 \rightarrow \tau_3) \quad \Gamma \vdash \mathbf{e}_2 : \tau_2$$

$$\Gamma \vdash \{\mathbf{e}_1 \ \mathbf{e}_2\} : \tau_3$$

$$\emptyset \vdash \{\text{fun } \{x : \text{bool}\} \{\text{if } x \ 1 \ 2\}\} : (\text{bool} \rightarrow \text{num}) \quad \emptyset \vdash \text{true} : \text{bool}$$

$$\emptyset \vdash \{\{\text{fun } \{x : \text{bool}\} \{\text{if } x \ 1 \ 2\}\} \ \text{true}\} : \text{num}$$

$$\emptyset \vdash \{\text{fun } \{x : \text{bool}\} \{\text{if } x \ 1 \ 2\}\} : (\text{bool} \rightarrow \text{num}) \quad \emptyset \vdash 5 : \text{num}$$

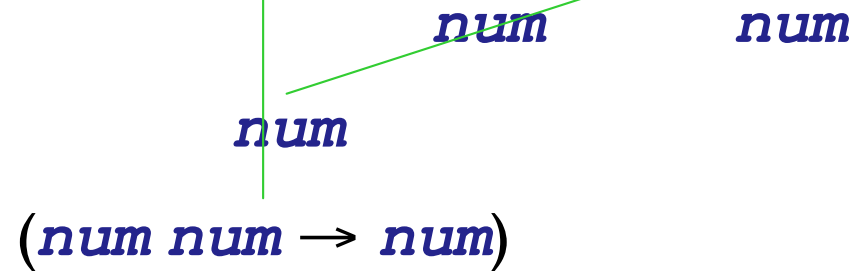
$$\emptyset \vdash \{\{\text{fun } \{x : \text{bool}\} \{\text{if } x \ 1 \ 2\}\} \ 5\} : \text{no type}$$

$$\emptyset \vdash 7 : \text{num} \quad \emptyset \vdash 5 : \text{num}$$

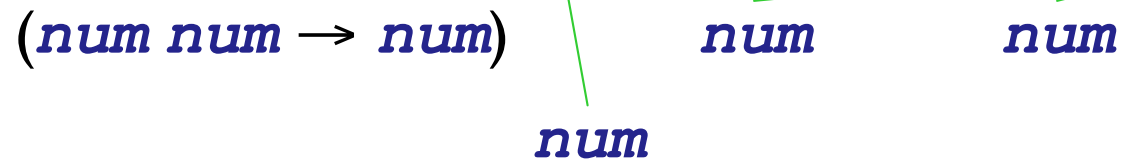
$$\emptyset \vdash \{7 \ 5\} : \text{no type}$$

Types: Multiple Arguments

`{fun {x : num y : num} {+ x y}}`



`{{fun {x : num y : num} {+ x y}} 5 6}`



`{{fun {x : num y : num} {+ x y}} 5}`



no type

Revised Function and Call Rules

$$\frac{\Gamma[\langle \text{id} \rangle_1 \leftarrow \tau_1 \dots \langle \text{id} \rangle_n \leftarrow \tau_n] \vdash \mathbf{e} : \tau_0}{\Gamma \vdash \{\text{fun } \langle \text{id} \rangle_1 : \tau_1 \dots \langle \text{id} \rangle_n : \tau_n \} \mathbf{e} : (\tau_1 \dots \tau_n \rightarrow \tau_0)}$$

$$\frac{\Gamma \vdash \mathbf{e}_0 : (\tau_1 \dots \tau_n \rightarrow \tau_0) \quad \Gamma \vdash \mathbf{e}_1 : \tau_1 \quad \dots \quad \Gamma \vdash \mathbf{e}_n : \tau_n}{\Gamma \vdash \{\mathbf{e}_0 \ \mathbf{e}_1 \ \dots \ \mathbf{e}_n\} : \tau_0}$$