

CS4613 Lecture 9: Types II

David Bremner

October 6, 2025

Function Application Example 2

$$\frac{\Gamma \vdash F : (T \rightarrow U) \quad \Gamma \vdash A : T}{\Gamma \vdash (F A) : U}$$

$$\frac{\Gamma \vdash (\text{lambda } (x : T) x) : (T \rightarrow ?) \quad \Gamma \vdash A : T}{\Gamma \vdash ((\text{lambda } (x : T) x) A) : ?}$$

$$\frac{\Gamma[x \leftarrow T] \vdash x : ?}{\Gamma \vdash (\text{lambda } (x : T) x) : (T \rightarrow ?)} \quad \Gamma \vdash A : T$$
$$\frac{}{\Gamma \vdash ((\text{lambda } (x : T) x) A) : ?}$$

Function Application Example x in plait

```
apply2 (define f (lambda (x) x))  
       (define v (f "hello world"))
```

Assume-Guarantee

Assume (given T will yield U). Guarantee argument has type T .

$$\frac{\Gamma \vdash F : (T \rightarrow U) \quad \Gamma \vdash A : T}{\Gamma \vdash (F \ A) : U}$$

Assume V has type T , guarantee type $(T \rightarrow U)$

p. 129

$$\frac{\Gamma[V \leftarrow T] \vdash \text{body} : U}{\Gamma \vdash (\text{lambda } (V : T) \text{ body}) : (T \rightarrow U)}$$

Function Application Example 3

Assume `odd?` yields `Bool` when given `Num`. Guarantee A has type `Num`.

$$\frac{\Gamma \vdash \text{odd?} : (\text{Num} \rightarrow \text{Bool}) \quad \Gamma \vdash A : \text{Num}}{\Gamma \vdash (\text{odd? } A) : ?}$$

Assume V has type `Num`, guarantee type $(\text{Num} \rightarrow \text{Bool})$

$$\frac{\Gamma[V \leftarrow \text{Num}] \vdash (\text{if } \dots \text{ true false}) : \text{Bool}}{\Gamma \vdash (\text{lambda } (V : \text{Num}) (\text{if } \dots \text{ true false})) : (\text{Num} \rightarrow \text{Bool})}$$

Infinite Loops

```
(let ([omega (lambda (x) (x x))])  
  (omega omega))
```

stacker

- ▶ substitution (and stacker) tells us this is an infinite loop.
- ▶ what type should it have?

└ Recursion

└ Infinite Loops

```
(let ([omega (lambda (x) (x x))])  
  (omega omega))
```

- ▶ substitution (and stacker) tells us this is an infinite loop.
- ▶ what type should it have?

1. For more information on this function (sometimes called the Y -combinator) see [The Little Schemer](#), or [The Why of Y](#)

Type rules and omega

```
(let ([omega (lambda (x) (x x))])  
  (omega omega))
```

stacker

$$\frac{\Gamma \vdash \text{omega} : (T \rightarrow U) \quad \Gamma \vdash \text{omega} : T}{\Gamma \vdash (\text{omega omega}) : U}$$

p. 129

- From our typing rule, we can derive the equation

$$T = (T \rightarrow U) \tag{1}$$

- But then we can substitute this equation into itself, forever.

Typing let1

$$\frac{\Gamma \vdash E : T \quad \Gamma[V \leftarrow T] \vdash B : U}{\Gamma \vdash (\text{let1 } V : T \text{ E } B) : U}$$

p. 130

- Do we *need* the annotation on V? Why or why not?

```
{lam {x : ?} x}
```

```
{let1 f : ? {lam {x : num} x} body}
```

A new kind of let



recmac

```
(define-syntax-rule (rec V : T E B)
  (local [(V : T) (define V E)] B))
```

```
#;(let ([fact (lambda (n)
                  (if (zero? n) 1 (* n (fact (- n 1))))))]
  (fact 10))
```

```
(rec fact : (Number -> Number)
  (lambda ([n : Number])
    (if (zero? n) 1 (* n (fact (- n 1)))))
  (fact 10))
```

Typing Rec

p. 132

$$\frac{? \quad \Gamma[V \leftarrow T] \vdash B : U}{\Gamma \vdash (\text{rec } V : T \text{ E } B) : U}$$

$$\frac{? \vdash E : T \quad \Gamma[V \leftarrow T] \vdash B : U}{\Gamma \vdash (\text{rec } V : T \text{ E } B) : U}$$

$$\frac{\Gamma[V \leftarrow T] \vdash E : T \quad \Gamma[V \leftarrow T] \vdash B : U}{\Gamma \vdash (\text{rec } V : T \text{ E } B) : U}$$

Typing rec example I

$$\frac{\Gamma[f \leftarrow T] \vdash (\text{lambda } (n : \text{Num}) \ 0) : T \quad \Gamma[f \leftarrow T] \vdash (f \ 0) : U}{\Gamma \vdash (\text{rec } f : T \ (\text{lambda } (n : \text{Num}) \ 0) \ (f \ 0)) : U}$$

- ▶ only one choice for T
- ▶ Γ does not matter

Typing rec example II

$$\frac{\Gamma[f \leftarrow T] \vdash (\text{lambda } (n : \text{Num}) \ n) : T \quad \Gamma[f \leftarrow T] \vdash (f \ 0) : U}{\Gamma \vdash (\text{rec } f : T \ (\text{lambda } (n : \text{Num}) \ n)(f \ 0)) : U}$$

- Still only one choice for T

Typing rec example III

$$\frac{\begin{array}{c} \Gamma[f \leftarrow T] \vdash (\text{lambda } (n : \text{Num}) (f \ n)) : T \\ \Gamma[f \leftarrow T] \vdash (f \ n) : U \end{array}}{\Gamma \vdash (\text{rec } f : T (\text{lambda } (n : \text{Num}) (f \ n))(f \ 0)) : U}$$

- ▶ What can we infer without Γ ?
- ▶ What does the type system not tell us here?

Typing factorial I/

```
(rec fact : (Number -> Number)
  (lambda ([n : Number])
    (if (zero? n) 1 (* n (fact (- n 1)))))
  (fact 10))
```

$$\frac{\begin{array}{c} \Gamma[\text{fact} \leftarrow (\text{Num} \rightarrow \text{Num})] \vdash (\text{lambda } \dots) : \text{Num} \rightarrow \text{Num} \\ \Gamma[\text{fact} \leftarrow (\text{Num} \rightarrow \text{Num})] \vdash (\text{fact } 10) : U \end{array}}{\Gamma \vdash (\text{rec fact} : (\text{Num} \rightarrow \text{Num}) (\text{lambda } \dots) (\text{fact } 10)) : U}$$

Typing factorial II/

$$\frac{\frac{\Gamma[\dots] \vdash (\text{lambda } \dots) : \text{Num} \rightarrow \text{Num} \quad \Gamma[\text{fact} \leftarrow (\text{Num} \rightarrow \text{Num})] \vdash \text{fact} : (W \rightarrow U) \quad \Gamma[\dots] \vdash 10 : W}{\Gamma[\text{fact} \leftarrow (\text{Num} \rightarrow \text{Num})] \vdash (\text{fact } 10) : U}}{\Gamma \vdash (\text{rec fact} : (\text{Num} \rightarrow \text{Num}) (\text{lambda } \dots) (\text{fact } 10)) : U}$$

$$W = U = \text{Num}$$

$$\frac{\Gamma[\text{fact} \leftarrow (\text{Num} \rightarrow \text{Num})] \vdash (\text{if } \dots (* \text{ n } (\text{fact } (- \text{ n } 1)))) : \text{Num}}{\Gamma[\text{fact} \leftarrow (\text{Num} \rightarrow \text{Num})] \vdash (\text{lambda } \dots) : (\text{Num} \rightarrow \text{Num})}$$

Typing factorial III/

$$\frac{\Gamma[\mathbf{fact} \leftarrow (\text{Num} \rightarrow \text{Num})] \vdash (\mathbf{if} \dots (* \ n \ (\mathbf{fact} \ (- \ n \ 1)))) : \text{Num}}{\Gamma[\mathbf{fact} \leftarrow (\text{Num} \rightarrow \text{Num})] \vdash (\mathbf{lambda} \dots) : (\text{Num} \rightarrow \text{Num})}$$
$$\frac{\frac{\dots \ \Gamma[\mathbf{fact} \leftarrow (\text{Num} \rightarrow \text{Num})] \vdash (\mathbf{fact} \ (- \ n \ 1)) : \text{Num}}{\Gamma[\mathbf{fact} \leftarrow (\text{Num} \rightarrow \text{Num})] \vdash (\mathbf{if} \dots (* \ n \ (\mathbf{fact} \ (- \ n \ 1)))) : \text{Num}}}{\Gamma[\mathbf{fact} \leftarrow (\text{Num} \rightarrow \text{Num})] \vdash (\mathbf{lambda} \dots) : (\text{Num} \rightarrow \text{Num})}$$