CS4613 Lecture 9: Types II

David Bremner

October 8, 2025

Function Application Example 2

$$\frac{\Gamma \vdash \texttt{F} : (T \to U) \qquad \Gamma \vdash \texttt{A} : T}{\Gamma \vdash \texttt{(F A)} : U}$$

$$\frac{\Gamma \vdash \texttt{(lambda (x : } T\texttt{) x)} : (T \to ?) \qquad \Gamma \vdash \texttt{A} : T}{\Gamma \vdash \texttt{((lambda (x : } T\texttt{) x) A)} : ?}$$

$$\frac{\dfrac{\Gamma[x \leftarrow T] \vdash \texttt{x} : ?}{\Gamma \vdash \texttt{(lambda (x : } T\texttt{) x)} : (T \to ?)} \qquad \Gamma \vdash \texttt{A} : T}{\Gamma \vdash \texttt{((lambda (x : } T\texttt{) x) A)} : ?}$$

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.

$$\frac{\Gamma \vdash \texttt{F} : (\mathsf{T} \rightarrow \mathsf{U}) \qquad \Gamma \vdash \texttt{A} : \mathsf{T}}{\Gamma \vdash \texttt{(F A)} : U}$$

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

p. 129

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

Function Application Example 3

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

$$\frac{\Gamma \vdash \texttt{odd?} : (\mathrm{Num} \to \mathrm{Bool}) \qquad \Gamma \vdash \texttt{A} : \mathrm{Num}}{\Gamma \vdash \texttt{(odd? A)} : ?}$$

Assume V has type Num, guarantee type $(\mathrm{Num} \to \mathrm{Bool})$

$$\frac{\Gamma[V \leftarrow \mathrm{Num}] \vdash \texttt{(if ... true false)} : \mathrm{Bool}}{\Gamma \vdash \texttt{(lambda (V : } \mathrm{Num}\texttt{) (if ... true false)} : (\mathrm{Num} \to \mathrm{Bool})}$$

Infinite Loops

stacker

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

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

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?

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

$$\frac{\Gamma \vdash \ \texttt{omega} \ : (T \to U) \qquad \Gamma \vdash \ \texttt{omega} \ : T}{\Gamma \vdash \ \texttt{(omega omega)} \ : U}$$

p. 129

- From our typing rule, we can derive the equation

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

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

Typing `let1`

p. 130

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

- 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

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

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

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

Typing `rec` example I

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

- only one choice for T
- Γ does not matter

Typing `rec` example II

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

- Still only one choice for T

Typing `rec` example III

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

- 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))
```

$$\frac{\Gamma[\texttt{fact} \leftarrow (\mathrm{Num} \to \mathrm{Num})] \vdash \texttt{(lambda ...)} : \mathrm{Num} \to \mathrm{Num} \qquad \Gamma[\texttt{fact} \leftarrow (\mathrm{Num} \to \mathrm{Num})] \vdash \texttt{(fact 10)} : U}{\Gamma \vdash \texttt{(rec fact : (}\mathrm{Num}{\to}\mathrm{Num}\texttt{) (lambda ...) (fact 10))} : U}$$

Typing factorial II/

$$\dfrac{\Gamma[\texttt{fact} \leftarrow (\mathrm{Num} \to \mathrm{Num})] \vdash \texttt{fact} : (W \to U) \quad \Gamma[\ldots] \vdash 10 : W}{\dfrac{\Gamma[\texttt{fact} \leftarrow (\mathrm{Num} \to \mathrm{Num})] \vdash \texttt{(fact 10)} : U}{\Gamma \vdash \texttt{(rec fact : (Num}{\to}\texttt{Num) (lambda ...) (fact 10))} : U}}$$

$$\Gamma[\ldots] \vdash \texttt{(lambda ...)} : \mathrm{Num} \to \mathrm{Num}$$

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

$$\dfrac{\Gamma[\texttt{fact} \leftarrow (\mathrm{Num} \to \mathrm{Num})] \vdash \texttt{(if ...(* n (fact (- n 1))))} : \mathrm{Num}}{\Gamma[\texttt{fact} \leftarrow (\mathrm{Num} \to \mathrm{Num})] \vdash \texttt{(lambda ...)} : (\mathrm{Num} \to \mathrm{Num})}$$

Typing factorial III/

$$\frac{\Gamma[\texttt{fact} \leftarrow (\mathrm{Num} \to \mathrm{Num})] \vdash \texttt{(if ...(* n (fact (- n 1))))} : \mathrm{Num}}{\Gamma[\texttt{fact} \leftarrow (\mathrm{Num} \to \mathrm{Num})] \vdash \texttt{(lambda ...)} : (\mathrm{Num} \to \mathrm{Num})}$$

$$\frac{\dfrac{\ldots\ \Gamma[\texttt{fact} \leftarrow (\mathrm{Num} \to \mathrm{Num})] \vdash \texttt{(fact (- n 1))} : \mathrm{Num}}{\Gamma[\texttt{fact} \leftarrow (\mathrm{Num} \to \mathrm{Num})] \vdash \texttt{(if ...(* n (fact (- n 1))))} : \mathrm{Num}}}{\Gamma[\texttt{fact} \leftarrow (\mathrm{Num} \to \mathrm{Num})] \vdash \texttt{(lambda ...)} : (\mathrm{Num} \to \mathrm{Num})}$$

- last example needs type binding for n