CS4613 Lecture 9: Types II

David Bremner

February 13, 2024

Arrow Types

p. 123

$$\frac{\vdash \texttt{F} : (? \rightarrow ?) \qquad \vdash \texttt{A} : ?}{\vdash (\texttt{F}\ \texttt{A}) : ?}$$

$$\frac{\vdash \texttt{F} : (T \rightarrow ?) \qquad \vdash \texttt{A} : T}{\vdash (\texttt{F}\ \texttt{A}) : ?}$$

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

Function Application Example 1

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

$$\frac{\vdash \texttt{(lambda (x : T) 3)} : (T \rightarrow ?) \qquad \vdash \texttt{A} : T}{\vdash \texttt{((lambda (x) 3) A)} : ?}$$

- We can't really type check function application with out type rules for function definitions

Type annotations for lambda

lamannote

```
(define f (lambda ([x : Number]) (+ x 1)))
(define g (lambda (x) (+ x 1))) ;; type inference
```

Function definitions

p. 127

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

Function Application Example 1 redux

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

$$\frac{\Gamma \vdash \texttt{(lambda (x : T) 3)} : (T \rightarrow ?) \qquad \Gamma \vdash \texttt{A} : T}{\Gamma \vdash \texttt{((lambda (x) 3) A)} : ?}$$

$$\frac{\dfrac{\Gamma[x \leftarrow T] \vdash \texttt{3} : ?}{\Gamma \vdash \texttt{(lambda (x : T) 3)} : (T \rightarrow ?)} \qquad \Gamma \vdash \texttt{A} : T}{\Gamma \vdash \texttt{((lambda (x) 3) A)} : ?}$$

Function Application Example 1 in plait

apply1

```
(define f (lambda (x) 3))
(define v (f "hello world"))
```

Function Application Example 2

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

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

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

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.

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

p. 128

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

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

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

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

p. 129

- From our typing rule, we can derive the equation

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

- 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}}
```

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) (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

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

Typing `rec` example II

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

Typing `rec` example III

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