```
"will eval to"
            5: Num
       +rue: Bool
        false: Bool
          3: Nu~
                                            TR-FALSE FAISC : Bool
                      TR-Num n: \mathsf{Num}
  general rule for typing, number expressions
                                because 3: Non and 7: Non
                                     and + produces Num when that's true
     (+ 37): Num

(+ true false): TYPE!
        (< 3 7): Bool
                                   because (+ 37): Num and 4: Num
                                       and + produces Nun....
    (+ (+ 3 7) 4): Num
  TR-PLUS \frac{\Gamma + e_1: \text{Nun}}{\Gamma + e_1 e_2: \text{Nun}} TR-Less
                                      It e: Non Prez: Non
(e:: Nun) 1 (ez: Nun) -> (+ e, ez): Nun
```

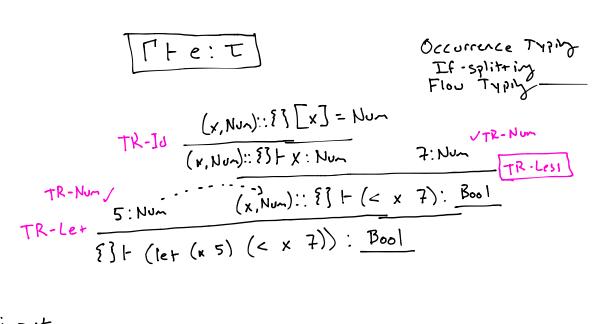
T:= {x: t, ... } [[a] = Num Tra: Non Maj = Bool Tha: Bool "add x env [" (x,t):: [C Pre: T (x,T):: Trez: Tz TR-ID $\Gamma \vdash (\mathsf{let} \ (x \ e_1) \ e_2) : left \mathsf{T}_{\mathsf{Z}}$ $\Gamma \vdash x :$ (let (x 5) (< 57) Pte:Bool Pte:T Pte:T

Pt (if e, e, e,): T $\begin{aligned} & \text{TR-IFA} \; \frac{\Gamma \vdash e_1 : \tau_1 \quad \Gamma \vdash e_2 : \tau_2 \quad \Gamma \vdash e_3 : \tau_3}{\Gamma \vdash \text{ (if } e_1 \ e_2 \ e_3) : \tau_1} \; & \text{TR-IFB} \; \frac{\Gamma \vdash e_1 : \text{Bool} \quad \Gamma \vdash e_2 : \tau_2 \quad \Gamma \vdash e_3 : \tau_3}{\Gamma \vdash \text{ (if } e_1 \ e_2 \ e_3) : \tau_2} \\ & \text{TR-IFC} \; \frac{\Gamma \vdash e_1 : \text{Bool} \quad \Gamma \vdash e_2 : \tau_2 \quad \Gamma \vdash e_3 : \tau_3}{\Gamma \vdash \text{ (if } e_1 \ e_2 \ e_3) : \tau_3} \end{aligned} \\ & \text{TR-IFD} \; \frac{\Gamma \vdash e_1 : T_1 \quad \Gamma \vdash e_2 : \tau_2 \quad \Gamma \vdash e_3 : \tau_3}{\Gamma \vdash \text{ (if } e_1 \ e_2 \ e_3) : \tau_2} \end{aligned}$ - Give result TzUT3 - Force Tz=T3 TR-IfE: None of the above (let (a (if (< 12) 3 4)) - Special rule for the/false Mre,: I, Mrez: Iz

Mrt(if false e, ez): Iz

Typed Racket

Sam Tobin-Hochstadt



input
"tre" number lit "
"falx"

(input, Num):: }} + (+ input 1):

(if input 3 false) Works w/o types

(an't work with these types