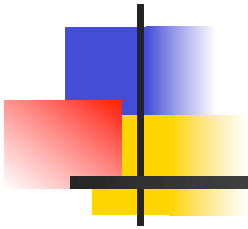


Programming Languages and Compilers (CS 421)

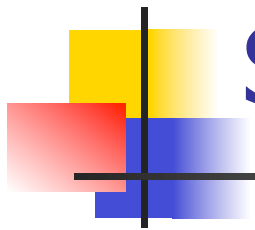


Elsa L Gunter

2112 SC, UIUC

<http://courses.engr.illinois.edu/cs421>

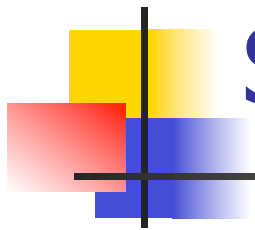
Based in part on slides by Mattox Beckman, as updated by Vikram Adve and Gul Agha



Simple Example

- Let $\Gamma = [x:\text{int} ; y:\text{bool}]$
- Show $\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}$
- Start building the proof tree from the bottom up

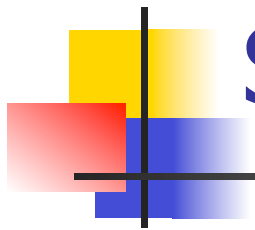
$$\frac{\quad ?}{\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}}$$



Simple Example

- Let $\Gamma = [x:\text{int} ; y:\text{bool}]$
- Show $\Gamma \vdash y \mid\mid (x + 3 > 6) : \text{bool}$
- Which rule has this as a conclusion?

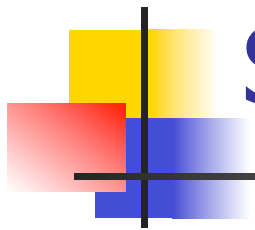
$$\frac{?}{\Gamma \vdash y \mid\mid (x + 3 > 6) : \text{bool}}$$



Simple Example

- Let $\Gamma = [x:\text{int} ; y:\text{bool}]$
- Show $\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}$
- Boolean Connectives: $\parallel$

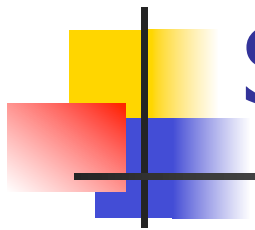
$$\frac{\Gamma \vdash y : \text{bool} \quad \Gamma \vdash x + 3 > 6 : \text{bool}}{\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}}$$



Simple Example

- Let $\Gamma = [x:\text{int} ; y:\text{bool}]$
- Show $\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}$
- Pick an assumption to prove

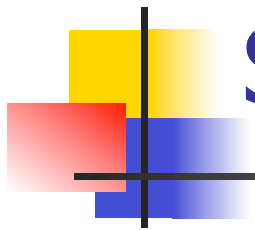
$$\frac{\frac{?}{\Gamma \vdash y : \text{bool}} \quad \Gamma \vdash x + 3 > 6 : \text{bool}}{\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}}$$



Simple Example

- Let $\Gamma = [x:\text{int} ; y:\text{bool}]$
- Show $\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}$
- Which rule has this as a conclusion?

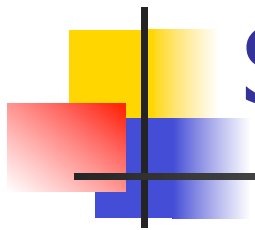
$$\frac{\frac{?}{\Gamma \vdash y : \text{bool}} \quad \Gamma \vdash x + 3 > 6 : \text{bool}}{\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}}$$



Simple Example

- Let $\Gamma = [x:\text{int} ; y:\text{bool}]$
- Show $\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}$
- Axiom for variables

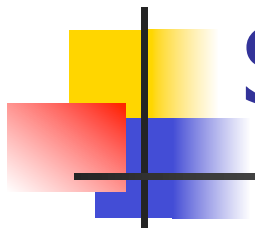
$$\frac{\Gamma \vdash y : \text{bool} \quad \Gamma \vdash x + 3 > 6 : \text{bool}}{\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}}$$



Simple Example

- Let $\Gamma = [x:\text{int} ; y:\text{bool}]$
- Show $\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}$
- Pick an assumption to prove

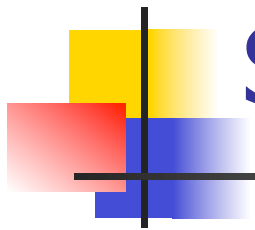
$$\frac{\frac{}{\Gamma \vdash y : \text{bool}} \quad \frac{\frac{}{\Gamma \vdash x + 3 > 6 : \text{bool}}{?}}{\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}}}{\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}}$$



Simple Example

- Let $\Gamma = [x:\text{int} ; y:\text{bool}]$
- Show $\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}$
- Which rule has this as a conclusion?

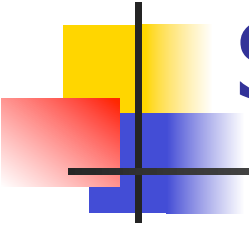
$$\frac{\frac{}{\Gamma \vdash y : \text{bool}} \quad \frac{\frac{}{\Gamma \vdash x + 3 > 6 : \text{bool}}{?}}{\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}}}{\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}}$$



Simple Example

- Let $\Gamma = [x:\text{int} ; y:\text{bool}]$
- Show $\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}$
- Arithmetic relations

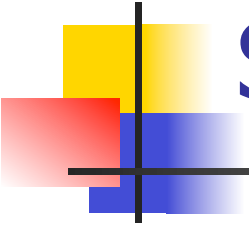
$$\frac{\frac{}{\Gamma \vdash y : \text{bool}} \quad \frac{\Gamma \vdash x + 3 : \text{int} \quad \Gamma \vdash 6 : \text{int}}{\Gamma \vdash x + 3 > 6 : \text{bool}}}{\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}}$$



Simple Example

- Let $\Gamma = [x:\text{int} ; y:\text{bool}]$
- Show $\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}$
- Pick an assumption to prove

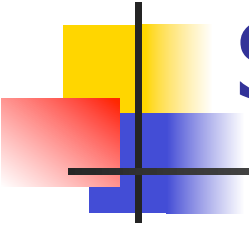
$$\frac{\frac{\Gamma \vdash y : \text{bool}}{\Gamma \vdash y : \text{bool}} \quad \frac{\Gamma \vdash x + 3 : \text{int} \quad \frac{\Gamma \vdash 6 : \text{int}}{?}}{\Gamma \vdash x + 3 > 6 : \text{bool}}}{\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}}$$



Simple Example

- Let $\Gamma = [x:\text{int} ; y:\text{bool}]$
- Show $\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}$
- Which rule has this as a conclusion?

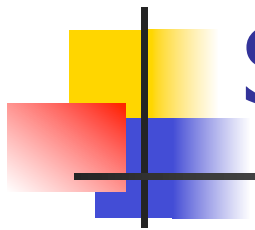
$$\frac{\frac{\Gamma \vdash y : \text{bool}}{\Gamma \vdash y : \text{bool}} \quad \frac{\Gamma \vdash x + 3 : \text{int} \quad \frac{\Gamma \vdash 6 : \text{int}}{?}}{\Gamma \vdash x + 3 > 6 : \text{bool}}}{\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}}$$



Simple Example

- Let $\Gamma = [x:\text{int} ; y:\text{bool}]$
- Show $\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}$
- Axiom for constants

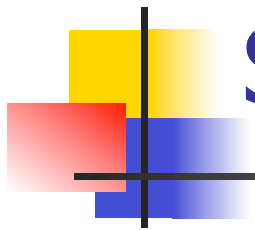
$$\frac{\frac{}{\Gamma \vdash y : \text{bool}} \quad \frac{\Gamma \vdash x + 3 : \text{int} \quad \overline{\Gamma \vdash 6 : \text{int}}}{\Gamma \vdash x + 3 > 6 : \text{bool}}}{\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}}$$



Simple Example

- Let $\Gamma = [x:\text{int} ; y:\text{bool}]$
- Show $\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}$
- Pick an assumption to prove

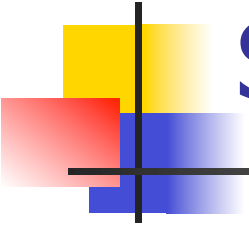
$$\begin{array}{c}
 \frac{}{\Gamma \vdash y : \text{bool}} \quad \frac{\frac{}{\Gamma \vdash x + 3 : \text{int}} \quad \frac{}{\Gamma \vdash 6 : \text{int}}}{\Gamma \vdash x + 3 > 6 : \text{bool}}}{\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}}
 \end{array}$$



Simple Example

- Let $\Gamma = [x:\text{int} ; y:\text{bool}]$
- Show $\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}$
- Which rule has this as a conclusion?

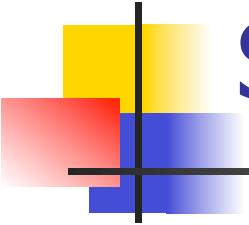
$$\begin{array}{c}
 \frac{}{\Gamma \vdash y : \text{bool}} \quad \frac{\frac{}{\Gamma \vdash x + 3 : \text{int}} \quad \frac{}{\Gamma \vdash 6 : \text{int}}}{\Gamma \vdash x + 3 > 6 : \text{bool}}}{\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}}
 \end{array}$$



Simple Example

- Let $\Gamma = [x:\text{int} ; y:\text{bool}]$
- Show $\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}$
- Arithmetic operations

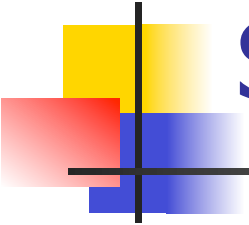
$$\frac{\frac{\Gamma \vdash x : \text{int} \quad \Gamma \vdash 3 : \text{int}}{\Gamma \vdash x + 3 : \text{int}} \quad \frac{}{\Gamma \vdash 6 : \text{int}}}{\frac{\Gamma \vdash y : \text{bool} \quad \Gamma \vdash x + 3 > 6 : \text{bool}}{\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}}}$$



Simple Example

- Let $\Gamma = [x:\text{int} ; y:\text{bool}]$
- Show $\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}$
- Pick an assumption to prove

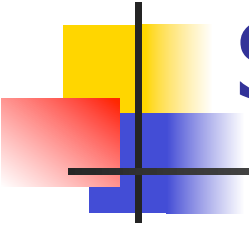
$$\frac{\frac{\frac{\Gamma \vdash x : \text{int}}{\Gamma \vdash x + 3 : \text{int}} \quad \frac{\Gamma \vdash 3 : \text{int}}{\Gamma \vdash 6 : \text{int}}}{\Gamma \vdash x + 3 > 6 : \text{bool}} \quad \Gamma \vdash y : \text{bool}}{\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}}$$



Simple Example

- Let $\Gamma = [x:\text{int} ; y:\text{bool}]$
- Show $\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}$
- Which rule has this as a conclusion?

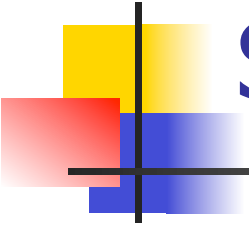
$$\begin{array}{c}
 \text{?} \\
 \hline
 \Gamma \vdash x : \text{int} \quad \Gamma \vdash 3 : \text{int} \\
 \hline
 \Gamma \vdash x + 3 : \text{int} \quad \Gamma \vdash 6 : \text{int} \\
 \hline
 \Gamma \vdash y : \text{bool} \quad \Gamma \vdash x + 3 > 6 : \text{bool} \\
 \hline
 \Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}
 \end{array}$$



Simple Example

- Let $\Gamma = [x:\text{int} ; y:\text{bool}]$
- Show $\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}$
- Axiom for constants

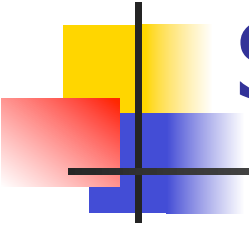
$$\frac{\frac{\Gamma \vdash x : \text{int} \quad \overline{\Gamma \vdash 3 : \text{int}}}{\Gamma \vdash x + 3 : \text{int}} \quad \overline{\Gamma \vdash 6 : \text{int}}}{\frac{\Gamma \vdash y : \text{bool} \quad \Gamma \vdash x + 3 > 6 : \text{bool}}{\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}}}$$



Simple Example

- Let $\Gamma = [x:\text{int} ; y:\text{bool}]$
- Show $\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}$
- Pick an assumption to prove

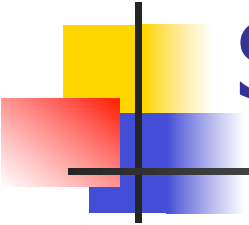
$$\frac{\frac{\frac{\frac{\Gamma \vdash x : \text{int}}{\Gamma \vdash x + 3 : \text{int}}}{\Gamma \vdash y : \text{bool}} \quad \frac{\frac{\frac{\Gamma \vdash 3 : \text{int}}{\Gamma \vdash 6 : \text{int}}}{\Gamma \vdash x + 3 > 6 : \text{bool}}}{\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}}}{\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}}$$



Simple Example

- Let $\Gamma = [x:\text{int} ; y:\text{bool}]$
- Show $\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}$
- Which rule has this as a conclusion?

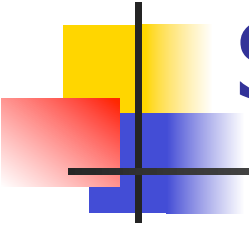
$$\frac{\frac{\frac{\Gamma \vdash x : \text{int}}{\Gamma \vdash x + 3 : \text{int}} \quad \frac{\Gamma \vdash 3 : \text{int}}{\Gamma \vdash 6 : \text{int}}}{\Gamma \vdash x + 3 > 6 : \text{bool}} \quad \Gamma \vdash y : \text{bool}}{\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}}$$



Simple Example

- Let $\Gamma = [x:\text{int} ; y:\text{bool}]$
- Show $\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}$
- Axiom for variables

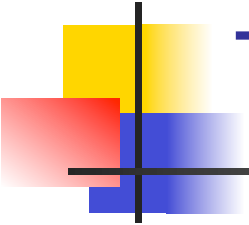
$$\frac{\frac{\frac{\Gamma \vdash x : \text{int}}{} \quad \frac{\Gamma \vdash 3 : \text{int}}{}}{\Gamma \vdash x + 3 : \text{int}} \quad \frac{\Gamma \vdash 6 : \text{int}}{} \quad \frac{\Gamma \vdash y : \text{bool} \quad \Gamma \vdash x + 3 > 6 : \text{bool}}{\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}}}{}$$



Simple Example

- Let $\Gamma = [x:\text{int} ; y:\text{bool}]$
- Show $\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}$
- No more assumptions! DONE!

$$\frac{\frac{\frac{\Gamma \vdash x : \text{int}}{} \quad \frac{\Gamma \vdash 3 : \text{int}}{}}{\Gamma \vdash x + 3 : \text{int}} \quad \frac{\Gamma \vdash 6 : \text{int}}{} \quad \frac{\Gamma \vdash y : \text{bool}}{} \quad \frac{\Gamma \vdash x + 3 > 6 : \text{bool}}{}}{\Gamma \vdash y \parallel (x + 3 > 6) : \text{bool}}$$

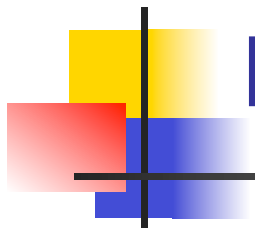


Type Variables in Rules

- If_then_else rule:

$$\frac{\Gamma \vdash e_1 : \text{bool} \quad \Gamma \vdash e_2 : \tau \quad \Gamma \vdash e_3 : \tau}{\Gamma \vdash (\text{if } e_1 \text{ then } e_2 \text{ else } e_3) : \tau}$$

- τ is a type variable (meta-variable)
- Can take any type at all
- All instances in a rule application must get same type
- Then branch, else branch and if_then_else must all have same type

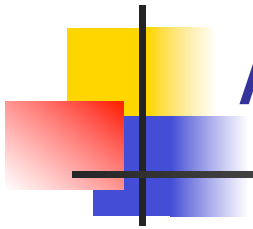


Function Application

- Application rule:

$$\frac{\Gamma \vdash e_1 : \tau_1 \rightarrow \tau_2 \quad \Gamma \vdash e_2 : \tau_1}{\Gamma \vdash (e_1 e_2) : \tau_2}$$

- If you have a function expression e_1 of type $\tau_1 \rightarrow \tau_2$ applied to an argument of type τ_1 , the resulting expression has type τ_2



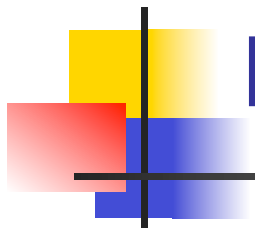
Application Examples

$$\frac{\Gamma \vdash \text{print_int} : \text{int} \rightarrow \text{unit} \quad \Gamma \vdash 5 : \text{int}}{\Gamma \vdash (\text{print_int } 5) : \text{unit}}$$

- $e_1 = \text{print_int}, e_2 = 5,$
- $\tau_1 = \text{int}, \tau_2 = \text{unit}$

$$\frac{\Gamma \vdash \text{map print_int} : \text{int list} \rightarrow \text{unit list} \quad \Gamma \vdash [3;7] : \text{int list}}{\Gamma \vdash (\text{map print_int } [3; 7]) : \text{unit list}}$$

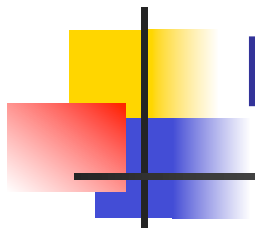
- $e_1 = \text{map print_int}, e_2 = [3; 7],$
- $\tau_1 = \text{int list}, \tau_2 = \text{unit list}$



Fun Rule

- Rules describe types, but also how the environment Γ may change
- Can only do what rule allows!
- fun rule:

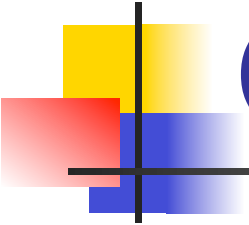
$$\frac{[x : \tau_1] + \Gamma \vdash e : \tau_2}{\Gamma \vdash \text{fun } x \rightarrow e : \tau_1 \rightarrow \tau_2}$$



Fun Examples

$$\frac{[y : \text{int}] + \Gamma \vdash y + 3 : \text{int}}{\Gamma \vdash \text{fun } y \rightarrow y + 3 : \text{int} \rightarrow \text{int}}$$

$$\frac{[f : \text{int} \rightarrow \text{bool}] + \Gamma \vdash f \ 2 :: [\text{true}] : \text{bool list}}{\Gamma \vdash (\text{fun } f \rightarrow f \ 2 :: [\text{true}]) : (\text{int} \rightarrow \text{bool}) \rightarrow \text{bool list}}$$



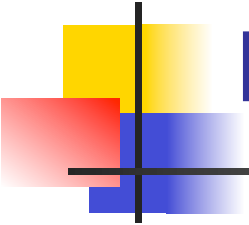
(Monomorphic) Let and Let Rec

- let rule:

$$\frac{\Gamma \vdash e_1 : \tau_1 \quad [x : \tau_1] + \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash (\text{let } x = e_1 \text{ in } e_2) : \tau_2}$$

- let rec rule:

$$\frac{[x : \tau_1] + \Gamma \vdash e_1 : \tau_1 \quad [x : \tau_1] + \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash (\text{let rec } x = e_1 \text{ in } e_2) : \tau_2}$$



Example (Assume Prim Op rule for ::)

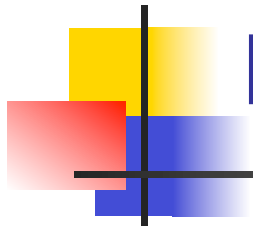
- Which rule do we apply?

?

| - (let rec one = 1 :: one in
let x = 2 in
fun y -> (x :: y :: one)) : int $\rightarrow$ int list



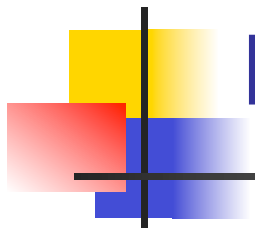
31



Proof of 1

- Which rule?

$[one : \text{int list}] \vdash (1 :: one) : \text{int list}$



Proof of 1

- Prim Op rule for $::$ (at int)

③

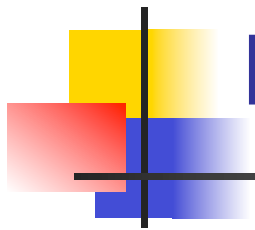
$[one : int\ list] \vdash 1 : int$

④

$[one : int\ list] \vdash$

$one : int\ list$

$[one : int\ list] \vdash (1 :: one) : int\ list$



Proof of 3

- Constant Rule

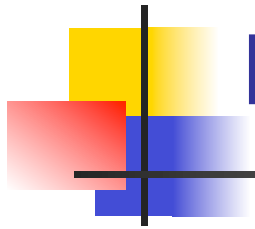
③

$$[one : \text{int list}] \vdash 1 : \text{int}$$

$$[one : \text{int list}] \vdash (1 :: one) : \text{int list}$$

④

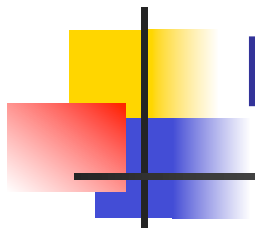
$$[one : \text{int list}] \vdash$$
$$one : \text{int list}$$



Proof of 4

- Rule for variables

$$\frac{}{[one : int\ list] \vdash one : int\ list}$$



Proof of 2

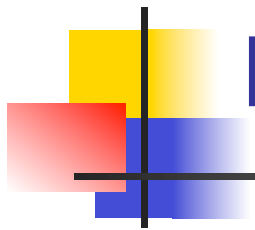
■ Constant

⑤ $[x:\text{int}; \text{one} : \text{int list}] \vdash$
 $\text{fun } y \rightarrow$

$(x :: y :: \text{one}))$

$[\text{one} : \text{int list}] \vdash 2:\text{int} \quad : \text{int} \rightarrow \text{int list}$

$[\text{one} : \text{int list}] \vdash (\text{let } x = 2 \text{ in}$
 $\text{fun } y \rightarrow (x :: y :: \text{one})) : \text{int} \rightarrow \text{int list}$

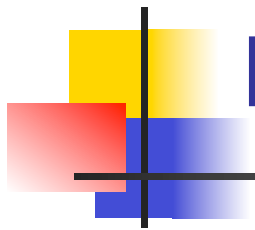


Proof of 5

Use function rule

?

$$\frac{}{[x:\text{int}; \text{one} : \text{int list}] \vdash \text{fun } y \rightarrow (x :: y :: \text{one})) \\ : \text{int} \rightarrow \text{int list}}$$

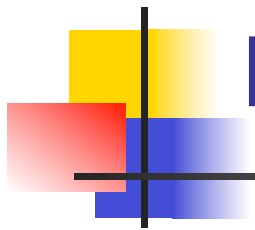


Proof of 5

Use Prim Op rule for ::

?

$$\frac{[y:\text{int}; x:\text{int}; \text{one} : \text{int list}] \vdash (x :: y :: \text{one}) : \text{int list}}{[x:\text{int}; \text{one} : \text{int list}] \vdash \text{fun } y \rightarrow (x :: y :: \text{one})) : \text{int} \rightarrow \text{int list}}$$



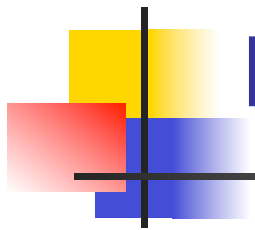
Proof of 5

Use Prim Op rule for ::

⑥

⑦

$$\frac{\begin{array}{c} [y:\text{int}; x:\text{int}; \text{one}:\text{int list}] \vdash x:\text{int} \\ [y:\text{int}; x:\text{int}; \text{one}:\text{int list}] \vdash (y :: \text{one}) : \text{int list} \end{array}}{\begin{array}{c} [y:\text{int}; x:\text{int}; \text{one} : \text{int list}] \vdash (x :: y :: \text{one}) : \text{int list} \\ [x:\text{int}; \text{one} : \text{int list}] \vdash \text{fun } y \rightarrow (x :: y :: \text{one}) \\ : \text{int} \rightarrow \text{int list} \end{array}}$$

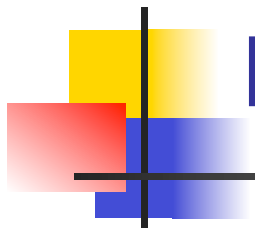


Proof of 6

⑥ Variable Rule

⑦

$$\frac{\begin{array}{c} [y:\text{int}; x:\text{int}; \text{one}:\text{int list}] \vdash x:\text{int} \\ [y:\text{int}; x:\text{int}; \text{one}:\text{int list}] \vdash (y :: \text{one}) : \text{int list} \end{array}}{\begin{array}{c} [y:\text{int}; x:\text{int}; \text{one} : \text{int list}] \vdash (x :: y :: \text{one}) : \text{int list} \\ [x:\text{int}; \text{one} : \text{int list}] \vdash \text{fun } y \rightarrow (x :: y :: \text{one}) \\ \quad : \text{int} \rightarrow \text{int list} \end{array}}$$



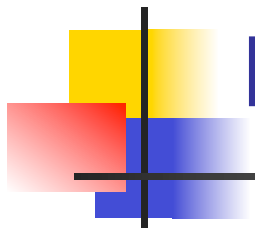
Proof of 7

Prim Op Rule ::

⑧

⑨

$$\frac{\begin{array}{c} [y:\text{int}; x:\text{int}; \text{one}:\text{int list}] \vdash [y:\text{int}; x:\text{int}; \text{one}:\text{int list}] \vdash \\ y:\text{int} \end{array} \quad \begin{array}{c} [y:\text{int}; x:\text{int}; \text{one}:\text{int list}] \vdash \\ \text{one} : \text{int list} \end{array}}{[y:\text{int}; x:\text{int}; \text{one} : \text{int list}] \vdash (y :: \text{one}): \text{int list}}$$



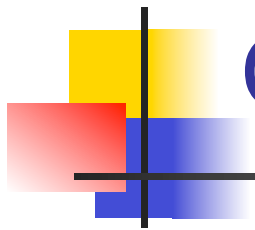
Proof of 8 and 9

Variable Rule

⑧

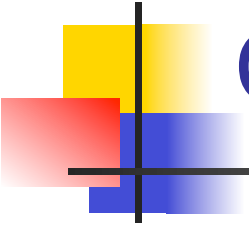
⑨

$$\frac{[y:\text{int}; x:\text{int}; \text{one}:\text{int list}] \vdash y:\text{int}}{[y:\text{int}; x:\text{int}; \text{one} : \text{int list}] \vdash (y :: \text{one}) : \text{int list}}$$
$$\frac{[y:\text{int}; x:\text{int}; \text{one}:\text{int list}] \vdash (y :: \text{one}) : \text{int list}}{[y:\text{int}; x:\text{int}; \text{one} : \text{int list}] \vdash (y :: \text{one}) : \text{int list}}$$



Curry - Howard Isomorphism

- Type Systems are logics; logics are type systems
- Types are propositions; propositions are types
- Terms are proofs; proofs are terms
- Functions space arrow corresponds to implication; application corresponds to modus ponens



Curry - Howard Isomorphism

- Modus Ponens

$$\frac{A \Rightarrow B \quad A}{B}$$

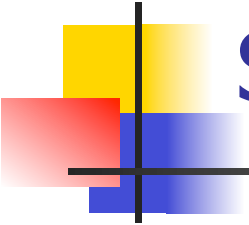
- Application

$$\frac{\Gamma \vdash e_1 : \alpha \rightarrow \beta \quad \Gamma \vdash e_2 : \alpha}{\Gamma \vdash (e_1 e_2) : \beta}$$



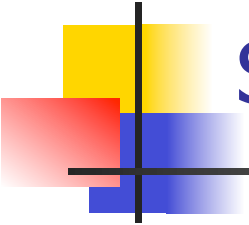
Mia Copa

- The above system can't handle polymorphism as in OCAML
- No type variables in type language (only meta-variable in the logic)
- Would need:
 - Object level type variables and some kind of type quantification
 - **let** and **let rec** rules to introduce polymorphism
 - Explicit rule to eliminate (instantiate) polymorphism



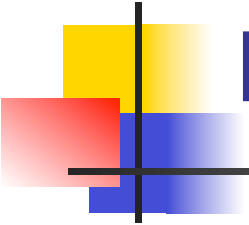
Support for Polymorphic Types

- Monomorphic Types (τ):
 - Basic Types: int, bool, float, string, unit, ...
 - Type Variables: $\alpha, \beta, \gamma, \delta, \epsilon$
 - Compound Types: $\alpha \rightarrow \beta$, int * string, bool list, ...
- Polymorphic Types:
 - Monomorphic types τ
 - Universally quantified monomorphic types
 - $\forall \alpha_1, \dots, \alpha_n. \tau$
 - Can think of τ as same as $\forall. \tau$



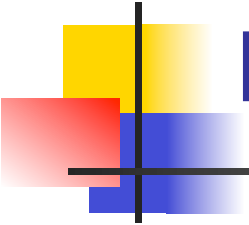
Support for Polymorphic Types

- Typing Environment Γ supplies polymorphic types (which will often just be monomorphic) for variables
- Free variables of monomorphic type just type variables that occur in it
 - Write $\text{FreeVars}(\tau)$
- Free variables of polymorphic type removes variables that are universally quantified
 - $\text{FreeVars}(\forall \alpha_1, \dots, \alpha_n . \tau) = \text{FreeVars}(\tau) - \{\alpha_1, \dots, \alpha_n\}$
- $\text{FreeVars}(\Gamma) =$ all FreeVars of types in range of Γ



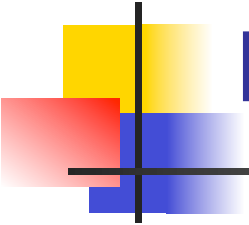
Monomorphic to Polymorphic

- Given:
 - type environment Γ
 - monomorphic type τ
 - τ shares type variables with Γ
- Want most polymorphic type for τ that doesn't break sharing type variables with Γ
- $\text{Gen}(\tau, \Gamma) = \forall \alpha_1, \dots, \alpha_n . \tau$ where
 $\{\alpha_1, \dots, \alpha_n\} = \text{freeVars}(\tau) - \text{freeVars}(\Gamma)$



Polymorphic Typing Rules

- A *type judgement* has the form
$$\Gamma \vdash \text{exp} : \tau$$
 - Γ uses polymorphic types
 - τ still monomorphic
- Most rules stay same (except use more general typing environments)
- Rules that change:
 - Variables
 - Let and Let Rec
 - Allow polymorphic constants
- Worth noting functions again



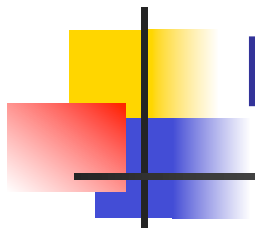
Polymorphic Let and Let Rec

- let rule:

$$\frac{\Gamma \vdash e_1 : \tau_1 \quad [x : \text{Gen}(\tau_1, \Gamma)] + \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash (\text{let } x = e_1 \text{ in } e_2) : \tau_2}$$

- let rec rule:

$$\frac{[x : \tau_1] + \Gamma \vdash e_1 : \tau_1 \quad [x : \text{Gen}(\tau_1, \Gamma)] + \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash (\text{let rec } x = e_1 \text{ in } e_2) : \tau_2}$$



Polymorphic Variables (Identifiers)

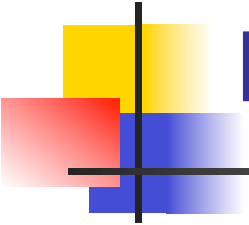
Variable axiom:

$$\overline{\Gamma \vdash x : \varphi(\tau)} \quad \text{if } \Gamma(x) = \forall \alpha_1, \dots, \alpha_n . \tau$$

- Where φ replaces all occurrences of $\alpha_1, \dots, \alpha_n$ by monotypes $\tau_1, \dots, \tau_n$
- Note: Monomorphic rule special case:

$$\overline{\Gamma \vdash x : \tau} \quad \text{if } \Gamma(x) = \tau$$

- Constants treated same way

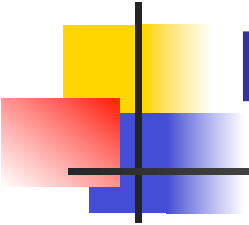


Fun Rule Stays the Same

- fun rule:

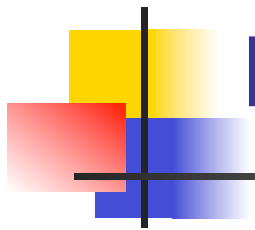
$$\frac{[x : \tau_1] + \Gamma \vdash e : \tau_2}{\Gamma \vdash \text{fun } x \rightarrow e : \tau_1 \rightarrow \tau_2}$$

- Types τ_1, τ_2 monomorphic
- Function argument must always be used at same type in function body



Polymorphic Example

- Assume additional constants:
- $\text{hd} : \forall \alpha. \alpha \text{ list} \rightarrow \alpha$
- $\text{tl} : \forall \alpha. \alpha \text{ list} \rightarrow \alpha \text{ list}$
- $\text{is_empty} : \forall \alpha. \alpha \text{ list} \rightarrow \text{bool}$
- $:: : \forall \alpha. \alpha \rightarrow \alpha \text{ list} \rightarrow \alpha \text{ list}$
- $[] : \forall \alpha. \alpha \text{ list}$

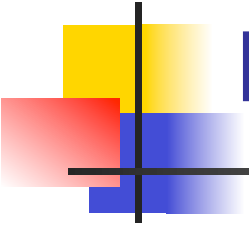


Polymorphic Example

- Show:

?

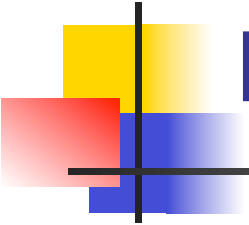
```
{ } |- let rec length =  
    fun l -> if is_empty l then 0  
             else 1 + length (tl l)  
in length ((::) 2 []) + length((::) true []) : int
```



Polymorphic Example: Let Rec Rule

■ Show: (1)	(2)
$\{\text{length} : \alpha \text{ list} \rightarrow \text{int}\}$	$\{\text{length} : \forall \alpha. \alpha \text{ list} \rightarrow \text{int}\}$
$\vdash \text{fun } l \rightarrow \dots$	$\vdash \text{length } ((::) 2 []) +$
$: \alpha \text{ list} \rightarrow \text{int}$	$\text{length}((::) \text{true } []) : \text{int}$

$\{\}$ $\vdash \text{let rec length} =$
 $\text{fun } l \rightarrow \text{if is_empty } l \text{ then } 0$
 $\text{else } 1 + \text{length } (\text{tl } l)$
in $\text{length } ((::) 2 []) + \text{length}((::) \text{true } []) : \text{int}$

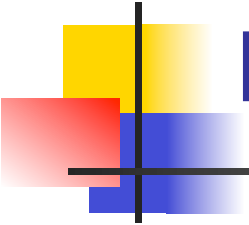


Polymorphic Example (1)

- Show:

?

```
{length:  $\alpha$  list -> int} |-  
fun l -> if is_empty l then 0  
        else 1 + length (tl l)  
:  
 $\alpha$  list -> int
```

Polymorphic Example (1): Fun Rule

■ Show: (3)

$\{\text{length} : \alpha \text{ list} \rightarrow \text{int}, \text{ l} : \alpha \text{ list} \} \vdash$

if is_empty l then 0

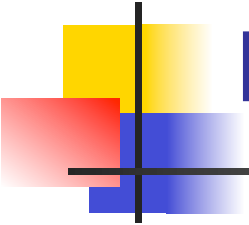
else length (hd l) + length (tl l) : int

$\{\text{length} : \alpha \text{ list} \rightarrow \text{int}\} \vdash$

fun l -> if is_empty l then 0

else 1 + length (tl l)

: $\alpha \text{ list} \rightarrow \text{int}$

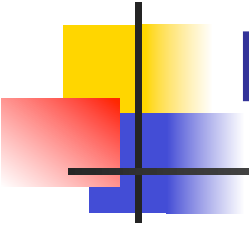


Polymorphic Example (3)

- Let $\Gamma = \{\text{length} : \alpha \text{ list} \rightarrow \text{int}, \text{ l} : \alpha \text{ list} \}$
- Show

?

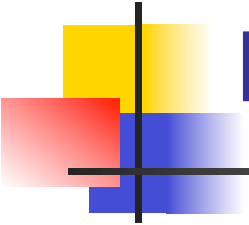
$\Gamma \vdash \text{if is_empty l then 0}$
 $\text{else } 1 + \text{length (tl l)} : \text{int}$



Polymorphic Example (3):IfThenElse

- Let $\Gamma = \{\text{length} : \alpha \text{ list} \rightarrow \text{int}, \text{ l} : \alpha \text{ list} \}$
- Show

$$\frac{\begin{array}{ccc} (4) & (5) & (6) \\ \Gamma \vdash \text{is_empty l} & \Gamma \vdash 0 : \text{int} & \Gamma \vdash 1 + \\ & : \text{bool} & \text{length (tl l)} : \text{int} \end{array}}{\Gamma \vdash \text{if is_empty l then 0} \\ \text{else } 1 + \text{length (tl l)} : \text{int}}$$

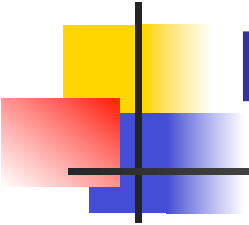


Polymorphic Example (4)

- Let $\Gamma = \{\text{length} : \alpha \text{ list} \rightarrow \text{int}, \text{ l} : \alpha \text{ list} \}$
- Show

?

$\Gamma \vdash \text{is_empty l} : \text{bool}$



Polymorphic Example (4):Application

- Let $\Gamma = \{\text{length} : \alpha \text{ list} \rightarrow \text{int}, \text{ l} : \alpha \text{ list} \}$
- Show

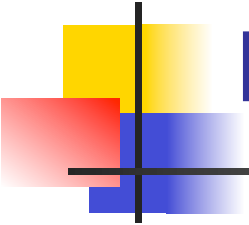
?

?

$$\Gamma \vdash \text{is_empty} : \alpha \text{ list} \rightarrow \text{bool}$$

$$\Gamma \vdash \text{l} : \alpha \text{ list}$$

$$\Gamma \vdash \text{is_empty l} : \text{bool}$$

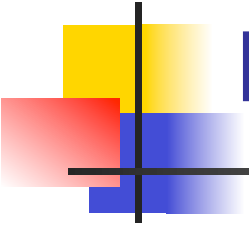


Polymorphic Example (4)

- Let $\Gamma = \{\text{length} : \alpha \text{ list} \rightarrow \text{int}, \text{ l} : \alpha \text{ list} \}$
- Show

By Const since $\alpha \text{ list} \rightarrow \text{bool}$ is
instance of $\forall \alpha. \alpha \text{ list} \rightarrow \text{bool}$?

$$\frac{}{\Gamma \vdash \text{is_empty} : \alpha \text{ list} \rightarrow \text{bool}}$$
$$\frac{}{\Gamma \vdash \text{l} : \alpha \text{ list}}$$
$$\Gamma \vdash \text{is_empty l} : \text{bool}$$



Polymorphic Example (4)

- Let $\Gamma = \{\text{length} : \alpha \text{ list} \rightarrow \text{int}, \text{ l} : \alpha \text{ list} \}$
- Show

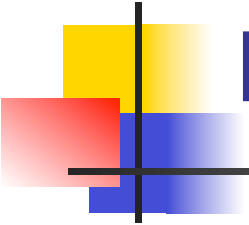
By Const since $\alpha \text{ list} \rightarrow \text{bool}$ is instance of $\forall \alpha. \alpha \text{ list} \rightarrow \text{bool}$ By Variable $\Gamma(\text{l}) = \alpha \text{ list}$

$\overline{\Gamma \vdash \text{is_empty} : \alpha \text{ list} \rightarrow \text{bool}}$

$\overline{\Gamma \vdash \text{l} : \alpha \text{ list}}$

$\Gamma \vdash \text{is_empty l} : \text{bool}$

- This finishes (4)



Polymorphic Example (5):Const

- Let $\Gamma = \{\text{length} : \alpha \text{ list} \rightarrow \text{int}, \text{ l} : \alpha \text{ list} \}$

- Show

By Const Rule

$$\frac{}{\Gamma \vdash 0 : \text{int}}$$

Polymorphic Example (6):Arith Op

- Let $\Gamma = \{\text{length} : \alpha \text{ list} \rightarrow \text{int}, \text{ l} : \alpha \text{ list} \}$
- Show

By Variable

$$\frac{}{\Gamma \vdash \text{length}} \quad (7)$$

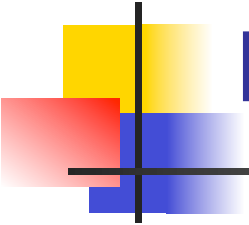
By Const

$$\frac{}{\Gamma \vdash 1 : \text{int}}$$

$$: \alpha \text{ list} \rightarrow \text{int} \quad \Gamma \vdash (\text{tl l}) : \alpha \text{ list}$$

$$\frac{}{\Gamma \vdash \text{length} (\text{tl l}) : \text{int}}$$

$$\frac{}{\Gamma \vdash 1 + \text{length} (\text{tl l}) : \text{int}}$$

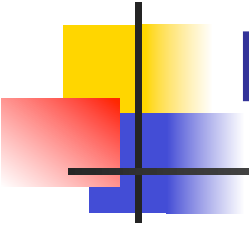


Polymorphic Example (7):App Rule

- Let $\Gamma = \{\text{length} : \alpha \text{ list} \rightarrow \text{int}, \text{ l} : \alpha \text{ list} \}$
- Show

By Const	By Variable
$\frac{}{\Gamma \vdash (\text{tl } l) : \alpha \text{ list} \rightarrow \alpha \text{ list}}$	$\frac{}{\Gamma \vdash l : \alpha \text{ list}}$
$\Gamma \vdash (\text{tl } l) : \alpha \text{ list}$	

By Const since $\alpha \text{ list} \rightarrow \alpha \text{ list}$ is instance of
 $\forall \alpha. \alpha \text{ list} \rightarrow \alpha \text{ list}$



Polymorphic Example: (2) by ArithOp

- Let $\Gamma' = \{\text{length} \forall \alpha. \alpha \text{ list} \rightarrow \text{int}\}$
- Show:

(8)

$\Gamma' \vdash$

$\text{length} ((::) 2 []) : \text{int}$

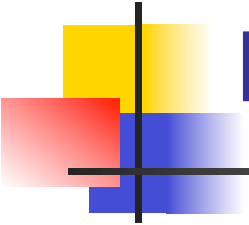
$\{\text{length}: \alpha. \alpha \text{ list} \rightarrow \text{int}\}$

$\vdash \text{length} ((::) 2 []) + \text{length} ((::) \text{true} []) : \text{int}$

(9)

$\Gamma' \vdash$

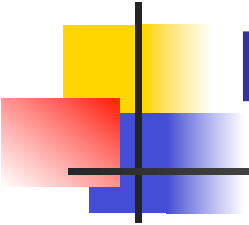
$\text{length} ((::) \text{true} []) : \text{int}$



Polymorphic Example: (8)AppRule

- Let $\Gamma' = \{\text{length} \forall \alpha. \alpha \text{ list} \rightarrow \text{int}\}$
- Show:

$$\frac{\Gamma' \vdash \text{length} : \text{int list} \rightarrow \text{int} \quad \Gamma' \vdash ((::)2 []): \text{int list}}{\text{list}}$$
$$\Gamma' \vdash \text{length } ((::)2 []) : \text{int}$$



Polymorphic Example: (8)AppRule

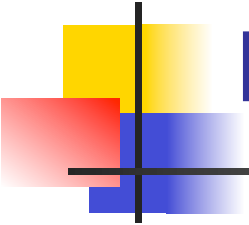
- Let $\Gamma' = \{\text{length} \forall \alpha. \alpha \text{ list} \rightarrow \text{int}\}$

- Show:

By Var since $\text{int list} \rightarrow \text{int}$ is instance of
 $\forall \alpha. \alpha \text{ list} \rightarrow \text{int}$

(10)

$$\frac{\Gamma' \vdash \text{length} : \text{int list} \rightarrow \text{int} \quad \Gamma' \vdash ((::)2 []):\text{int}}{\text{list}}$$
$$\Gamma' \vdash \text{length } ((::)2 []) : \text{int}$$

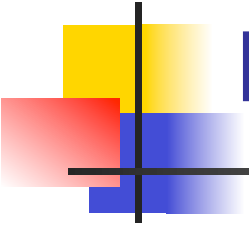


Polymorphic Example: (10)AppRule

- Let $\Gamma' = \{\text{length} \forall \alpha. \alpha \text{ list} \rightarrow \text{int}\}$
- Show:
- By Const since $\alpha \text{ list}$ is instance of $\forall \alpha. \alpha \text{ list}$

(11)

$$\frac{\Gamma' \vdash ((::) 2) : \text{int list} \rightarrow \text{int list} \quad \overline{\Gamma' \vdash [] : \text{int list}}}{\Gamma' \vdash ((::) 2 []) : \text{int list}}$$

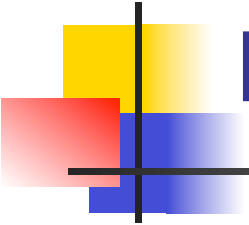


Polymorphic Example: (11)AppRule

- Let $\Gamma' = \{\text{length} \forall \alpha. \alpha \text{ list} \rightarrow \text{int}\}$
- Show:
- By Const since $\alpha \text{ list}$ is instance of

$$\frac{\forall \alpha. \alpha \text{ list}}{\Gamma' \vdash (::) : \text{int} \rightarrow \text{int list} \rightarrow \text{int list}} \quad \frac{\text{By Const}}{\Gamma' \vdash 2 : \text{int}}$$

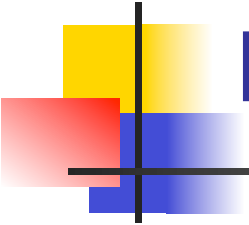
$$\Gamma' \vdash ((::) 2) : \text{int list} \rightarrow \text{int list}$$



Polymorphic Example: (9)AppRule

- Let $\Gamma' = \{\text{length} \forall \alpha. \alpha \text{ list} \rightarrow \text{int}\}$
- Show:

$$\frac{\begin{array}{c} \Gamma' \vdash \\ \text{length} : \text{bool list} \rightarrow \text{int} \end{array} \quad \begin{array}{c} \Gamma' \vdash \\ ((::) \text{ true } []) : \text{bool list} \end{array}}{\Gamma' \vdash \text{length } ((::) \text{ true } []) : \text{int}}$$



Polymorphic Example: (9)AppRule

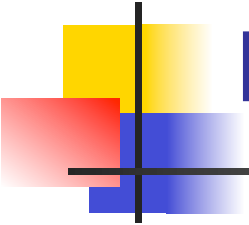
- Let $\Gamma' = \{\text{length} \forall \alpha. \alpha \text{ list} \rightarrow \text{int}\}$

- Show:

By Var since $\text{bool list} \rightarrow \text{int}$ is instance of
 $\forall \alpha. \alpha \text{ list} \rightarrow \text{int}$

(12)

$$\frac{\Gamma' \vdash \text{length} : \text{bool list} \rightarrow \text{int} \quad \Gamma' \vdash ((::) \text{ true } []) : \text{bool list}}{\Gamma' \vdash \text{length } ((::) \text{ true } []) : \text{int}}$$



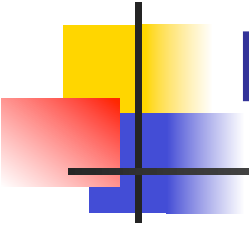
Polymorphic Example: (12)AppRule

- Let $\Gamma' = \{\text{length} \forall \alpha. \alpha \text{ list} \rightarrow \text{int}\}$
- Show:
- By Const since $\alpha \text{ list}$ is instance of $\forall \alpha. \alpha \text{ list}$

(13)

$$\frac{\Gamma' \vdash ((::)\text{true}) : \text{bool list} \rightarrow \text{bool list} \quad \overline{\Gamma' \vdash [] : \text{bool list}}}{\Gamma' \vdash ((::)\text{true} []) : \text{bool list}}$$

$\Gamma' \vdash ((::)\text{true} []) : \text{bool list}$



Polymorphic Example: (13)AppRule

- Let $\Gamma' = \{\text{length} \forall \alpha. \alpha \text{ list} \rightarrow \text{int}\}$
- Show:

By Const since bool list
is instance of $\forall \alpha. \alpha \text{ list}$

By Const

$\frac{\Gamma' \vdash}{(\::)\::\text{bool} \rightarrow \text{bool list} \rightarrow \text{bool list}}$	$\frac{\text{By Const}}{\Gamma' \vdash \text{true} : \text{bool}}$
$\Gamma' \vdash ((\::)\::) \text{true} : \text{bool list} \rightarrow \text{bool list}$	