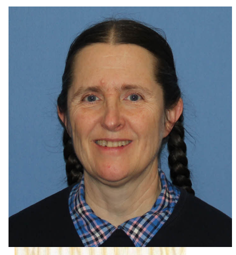
Programming Languages and Compilers (CS 421)

Elsa L Gunter

2112 SC, UIUC



https://courses.engr.illinois.edu/cs421/sp2023

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

Polymorphic Example

- Assume additional constants and mondaic and binary operators:
- hd : $\forall\alpha.\ \alpha$ list -> α (monadic)
- tl: $\forall\alpha.\ \alpha$ list -> α list (monadic)
- is_empty : $\forall\alpha.\ \alpha$ list -> bool (monadic)
- (::) : $\forall\alpha.\ \alpha$ -> α list -> α list (binary)
- [] : $\forall\alpha.\ \alpha$ list (constant)

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)

$$\frac{\{length:\alpha\ list \rightarrow int\} \vdash fun\ l \rightarrow \ldots : \alpha\ list \rightarrow int \qquad \{length:\forall\alpha.\ \alpha\ list \rightarrow int\} \vdash length\ (2 :: []) + length(true :: []) : int}{\{\} \vdash let\ rec\ length = fun\ l \rightarrow if\ is\_empty\ l\ then\ 0\ else\ 1 + length\ (tl\ l)\ in\ length\ (2 :: []) + length(true :: []) : int}$$

Polymorphic Example (1)

- Show:

?

{length:α list -> int} |-

fun l -> if is_empty l then 0

else 1 + length (tl l)

: α list -> int

Polymorphic Example (1): Fun Rule

- Show: (3)

$$\frac{\{length:\alpha\ list \rightarrow int,\ \ l:\alpha\ list\}\ |\text{-}\ \text{if is\_empty } l \text{ then } 0 \text{ else } 1 + length\ (tl\ l)\ :\ int}{\{length:\alpha\ list \rightarrow int\}\ |\text{-}\ \text{fun } l \rightarrow \text{if is\_empty } l \text{ then } 0 \text{ else } 1 + length\ (tl\ l)\ :\ \alpha\ list \rightarrow int}$$

Polymorphic Example (3)

- Let Γ ={length:α list -> int, l: α list }
- Show

?

Γ|- if is_empty l then 0
else 1 + length (tl l) : int

Polymorphic Example (3):IfThenElse

- Let Γ ={length:α list -> int, l: α list }
- Show

$$\dfrac{\dfrac{(4)}{\Gamma \mid\!- \text{ is\_empty l : bool}} \quad \dfrac{(5)}{\Gamma \mid\!- \text{ 0:int}} \quad \dfrac{(6)}{\Gamma \mid\!- \text{ 1 + length (tl l) : int}}}{\Gamma \mid\!- \text{ if is\_empty l then 0 else 1 + length (tl l) : int}}$$

Polymorphic Example (4)

- Let Γ ={length:α list -> int, l: α list }
- Show

$$\frac{?}{\Gamma \text{ |- is\_empty l : bool}}$$

Polymorphic Example (4):MonOpAP

- Let Γ ={length:α list -> int, l: α list }
- Show

By MonOpApp since α list -> bool is instance $\{\alpha \rightarrow \alpha\}$ of the type of is_empty: $\forall\alpha.$ α list -> bool

$$\dfrac{\dfrac{?}{\Gamma \mid\!-\ l : \alpha\ \text{list}}}{\Gamma \mid\!-\ \text{is\_empty}\ l : \text{bool}}$$

Polymorphic Example (4)

- Let Γ ={length:α list -> int, l: α list }
- Show

By Var

$$\frac{\frac{\text{Var}}{\Gamma \text{|- l : } \alpha \text{ list}}}{\Gamma \text{|- is\_empty l : bool}}$$

- This finishes (4)

Polymorphic Example (5):Const

- Let Γ ={length:α list -> int, l: α list }
- Show

By Const Rule

$$\frac{}{\Gamma \mid\!- \ 0:\text{int}}$$

Polymorphic Example (6): BinOp

- Let Γ ={length:α list -> int, l: α list }
- Show

By Variable
―――――――――――――
Γ|- length : α list -> int

(7)
―――――――――――――
Γ|- (tl l) : α list

By Const
―――――――――――――
Γ|- 1:int

App
―――――――――――――
Γ|- length (tl l) : int

―――――――――――――
Γ|- 1 + length (tl l) : int

Polymorphic Example (7): MonOp

- Let Γ ={length:α list -> int, l: α list }
- Show

By MonOpApp since α list -> α list is instance $\{\alpha \rightarrow \alpha\}$ of the type of tl: $\forall\alpha.\ \alpha$ list -> α list

$$\dfrac{\dfrac{?}{\Gamma \mid\!-\ l : \alpha\ \text{list}}}{\Gamma \mid\!-\ \text{tl}\ l : \text{bool}}$$

Polymorphic Example: (2) by BinOp

- Let Γ' = {length:$\forall\alpha.\ \alpha$ list -> int}
- Show:

$$\frac{\dfrac{(8)}{\Gamma' \ \vert\text{-}\ \text{length (2 :: []) :int}} \qquad \dfrac{(9)}{\Gamma' \ \vert\text{-}\ \text{length(true :: []) : int}}}{\{\text{length}:\forall\alpha.\ \alpha\ \text{list -> int}\}\ \vert\text{-}\ \text{length (2 :: []) + length(true :: []) : int}}$$

Polymorphic Example: (8)AppRule

- Let Γ' = {length:$\forall\alpha.\ \alpha$ list -> int}
- Show:

$$\frac{\dfrac{?}{\Gamma' \mid\!-\ \text{length : int list ->int}} \quad \dfrac{?}{\Gamma' \mid\!-\ \text{(2 :: []):int list}}}{\Gamma' \mid\!-\ \text{length (2 :: []) :int}}$$

Polymorphic Example: (8)AppRule

- Let Γ' = {length: $\forall\alpha.\ \alpha$ list -> int}
- Show:

By Var since int list -> int is instance $\{\alpha \rightarrow \text{int}\}$ of $\forall\alpha.\ \alpha$ list -> int (by $\{\alpha \rightarrow \text{int}\}$)

$$\frac{\dfrac{\text{Var}}{\Gamma' \mid\!- \text{length : int list -> int}} \qquad \dfrac{(10)}{\Gamma' \mid\!- (2 :: []) : \text{int list}}}{\Gamma' \mid\!- \text{length } (2 :: []) : \text{int}}$$

Polymorphic Example: (10)BinOpRule

- Let Γ' = {length:$\forall\alpha.\ \alpha$ list -> int}
- Show:
- By BinOp since :: has type

$\forall\alpha.\ \alpha$ -> α list -> α list (by {α ➔ int}

$$\dfrac{\dfrac{\text{Const}}{\Gamma' \mid\!- 2 : \text{int}} \qquad \dfrac{?}{\Gamma' \mid\!- [\,] : \text{int list}}}{\Gamma' \mid\!- (2 :: [\,]) : \text{int list}}$$

Polymorphic Example: (10)BinOpRule

- Let Γ' = {length:$\forall\alpha.\ \alpha$ list -> int}
- Show:
- By Const since int list is instance of $\forall\alpha.\ \alpha$ list (by {α ➔ int})

$$\dfrac{\dfrac{\text{Const}}{\Gamma' \text{ |- } 2 : \text{int}} \qquad \dfrac{\text{Const}}{\Gamma' \text{ |- } [\,] : \text{int list}}}{\Gamma' \text{ |- } (2 :: [\,]) : \text{int list}}$$

Polymorphic Example: (9)AppRule

- Let Γ' = {length:∀ α. α list -> int}
- Show:

$$\frac{\dfrac{?}{\Gamma' \text{ |- length :bool list ->int}} \qquad \dfrac{?}{\Gamma' \text{ |- (true :: []) :bool list}}}{\Gamma' \text{ |- length (true :: []) :int}}$$

Polymorphic Example: (9)AppRule

- Let Γ' = {length:∀ α. α list -> int}
- Show:
- Var since bool list -> int is instance of ∀α. α list -> int (by {α ➔ bool})

$$\dfrac{\dfrac{}{\Gamma' \mid\!- \text{ length} : \text{bool list} \to \text{int}} \qquad \dfrac{(11)}{\Gamma' \mid\!- (\text{true} :: []) : \text{bool list}}}{\Gamma' \mid\!- \text{ length (true} :: []) : \text{int}}$$

Polymorphic Example: (11)BinOpRule

- Let Γ' = {length: $\forall\alpha.\ \alpha$ list -> int}
- Show:
- By BinOp since :: has type

  $\forall\alpha.\ \alpha$ -> α list -> α list (by {α ➔ bool})

$$\frac{\dfrac{\text{Const}}{\Gamma' \text{ |- true : bool}} \qquad \dfrac{?}{\Gamma' \text{ |- [] : bool list}}}{\Gamma' \text{ |- (true :: []) : bool list}}$$

Polymorphic Example: (10)BinOpRule

- Let Γ' = {length:∀α. α list -> int}
- Show:
- By Const since bool list is instance of ∀α. α list (by {α ➔ bool})

$$\dfrac{\dfrac{\text{Const}}{\Gamma' \mid\!- \text{true} : \text{bool}} \qquad \dfrac{\text{Const}}{\Gamma' \mid\!- [\,] : \text{bool list}}}{\Gamma' \mid\!- (\text{true} :: [\,]) : \text{bool list}}$$