QuB

A Resource Aware Functional Programming Language

Apoorv Ingle

The University of Kansas

Table of Contents

Introduction and Motivation

- Hard problem: Resource management in evolving production code
- Resources: Files, database connections, anything that represents a shared state in the program

Resource Management: File Handling

- Modified File Handling API in Haskell

$$
\begin{aligned}
\texttt{openFile} &:: \texttt{FilePath} \rightarrow \texttt{IO FileHandle} \\
\texttt{closeFile} &:: \texttt{FileHandle} \rightarrow \texttt{IO} \; () \\
\texttt{readLine} &:: \texttt{FileHandle} \rightarrow \texttt{IO} \; (\texttt{String}, \texttt{FileHandle}) \\
\texttt{writeFile} &:: \texttt{String} \rightarrow \texttt{FileHandle} \\
&\quad \rightarrow \texttt{IO} \; ((), \; \texttt{FileHandle}) \\
\\
\texttt{upper} &:: \texttt{String} \rightarrow \texttt{String}
\end{aligned}
$$

Resource Management: File Handling

- File Handling in Haskell

```
do f      ← openFile "sample.txt"
   (s, f)  ← readLine f
   let c = upper s
   ((), f) ← writeLine f c
             .
             .
             .
   ()     ← closeFile f
```

Resource Management: File Handling

- File Handling in Haskell Gone Wrong (Part I)

```
do f      ← openFile "sample.txt"
   (s, f)  ← readLine f
   let c = upper s
   ((), f) ← writeLine f c
         .
         .
         .
   ()   ← closeFile f
         .
         .
         .
   ()   ← closeFile f
   return c
```

Resource Management: File Handling

- File Handling in Haskell Gone Wrong (Part I)

```
do f ← openFile "sample.txt"
   (s, f) ← readLine f
   let c = upper s
   ((), f) ← writeLine f c
        .
        .
        .
   () ← closeFile f
        .
        .
        .
   () ← closeFile f
   return c
```

- File is closed twice: Run time crash

Resource Management: File Handling

- File Handling in Haskell Gone Wrong (Part II)

```
do f     ← openFile "sample.txt"
   (s, f)   ← readLine f
   let c = upper s
   ((), f) ← writeLine f c
       .
       .
       .
   return c
```

- File Handling in Haskell Gone Wrong (Part II)

```
do f      ← openFile "sample.txt"
   (s, f)  ← readLine f
   let c = upper s
   ((), f) ← writeLine f c
      .
      .
      .
   return c /*File not closed!!*/
```

- File not closed: Memory leak

Resource Management: Exception Handling

- MonadError(?) in Haskell

```
class Monad m ⇒ MonadError e m | m → e where
    throwError :: e → m a
    catchError :: m a → (e → m a) → m a
```

- throwError starts exception processing
- catchError exception handler

- Using `MonadError` in Haskell

```
do f ← openFile "sample.txt"
   ((s, f)  ← readLine f
   let c = upper s
   ()  ← closeFile f
   return $ Right c) `catchError` (\_ →
                     return $ Left "Error in reading file")
```

- Execution path for exception file not closed Memory leak

Introduction and Motivation

'Well typed programs do not go wrong.' Robin Milner

Introduction and Motivation

'Well typed programs do not go wrong.' Robin Milner

- Can we do better? Can types guide us?

$\boxed{\text{Types and Typing Context}}$

$$
\begin{aligned}
t, u &\in \text{Type Variables} \\
\text{Types} \quad \tau, \upsilon &::= t \mid \iota \mid \tau \to \tau \\
\text{Typing Scheme} \quad \sigma &::= \tau \mid \forall t.\tau \\
\text{Typing Context} \quad \Gamma &::= \epsilon \mid \Gamma, x : \sigma
\end{aligned}
$$

$\boxed{\text{Language}}$

$$
\begin{aligned}
\text{Expressions} \quad M, N ::= {} & x \\
& \mid \lambda x.M \mid MN \\
& \mid \texttt{let } x = M \texttt{ in } N
\end{aligned}
$$

Background Work: Hindley-Milner Type System

Hindley-Milner (***HM***) type system

Type inferencing and Type checking:

- Algorithm $\mathcal{M}$(?)
- Algorithm $\mathcal{W}$(?)

Type unification:

- Robinson's Algorithm $\mathcal{U}$(?)

Background Work: Typing Rules STLC

$$\frac{x : \sigma \in \Gamma}{\Gamma \vdash x : \sigma}\ [\text{VAR}] \qquad \frac{\Gamma \vdash M : \sigma \qquad \Gamma_x, x : \sigma \vdash N : \tau}{\Gamma \vdash (\texttt{let}\ x = M\ \texttt{in}\ N) : \tau}\ [\text{LET}]$$

$$\frac{\Gamma \vdash M : \sigma \qquad t \notin \texttt{fvs}(\Gamma)}{\Gamma \vdash M : \forall t.\sigma}\ [\forall\ \text{I}] \qquad \frac{\Gamma \vdash M : \sigma \qquad (\sigma' \sqsubseteq \sigma)}{\Gamma \vdash M : \sigma'}\ [\forall\ \text{E}]$$

$$\frac{\Gamma_x, x : \tau \vdash M : \tau'}{\Gamma \vdash \lambda x.M : \tau \to \tau'}\ [\to\ \text{I}] \qquad \frac{\Gamma \vdash M : \tau \to \tau' \qquad \Gamma \vdash N : \tau}{\Gamma \vdash MN : \tau'}\ [\to\ \text{E}]$$

Background Work: Algorithm $\mathcal{W}$

$$\boxed{\mathcal{M}(\Gamma \vdash M : \tau) = S}$$

$$\mathcal{M}(\Gamma \vdash x : \tau) = \mathcal{U}(\tau, [\vec{u}/\vec{t}]\upsilon)$$
$$\text{where} \qquad \forall \vec{t}.\upsilon = \Gamma(x)$$

$$\mathcal{M}(\Gamma \vdash \lambda x.M : \tau) = S \circ S'$$
$$\text{where} \qquad S = \mathcal{U}(\tau, u_1 \to u_2)$$
$$S' = \mathcal{M}(S\Gamma, x : Su_1 \vdash M : Su_2)$$

$$\mathcal{M}(\Gamma \vdash MN : \tau) = S \circ S'$$
$$\text{where} \qquad S = \mathcal{M}(\Gamma \vdash M : u \to \tau)$$
$$S' = \mathcal{M}(S\Gamma \vdash N : Su)$$

$$\mathcal{M}(\Gamma \vdash (\texttt{let } x = M \texttt{ in } N) : \tau) = S \circ S'$$
$$\text{where} \qquad S = \mathcal{M}(\Gamma \vdash M : u)$$
$$\sigma = \texttt{Gen}(S\Gamma, Su)$$
$$S' = \mathcal{M}(S\Gamma, x : \sigma \vdash N : \tau)$$

Background Work: Curry-Howard Correspondence

- Types are Propostions
- Programs are Proofs

$$\textit{\textbf{HM}} \text{ type system} \equiv \text{Second Order Propositional Logic}$$

Language:

$$\text{Propostions \& connectives} \quad A, B, C ::= x \mid A \supset B \mid \forall x.B \mid \ldots$$
$$\text{Context} \quad \Gamma, \Delta ::= \epsilon \mid \Gamma, A$$

Logic Rules:

$$\frac{}{A \vdash A}\ [\mathrm{Ax}]$$

$$\frac{\Gamma \vdash B \qquad x \notin \Gamma}{\forall x.B}\ [\forall \mathrm{I}] \qquad \frac{\Gamma \vdash \forall x.B \qquad \Gamma \vdash A}{B[x/A]}\ [\forall \mathrm{E}]$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \supset B}\ [\supset \mathrm{I}] \qquad \frac{\Gamma \vdash A \supset B \qquad \Gamma \vdash A}{\Gamma \vdash B}\ [\supset \mathrm{E}]$$

Background Work: Substructural Logic

- Two implicit rules in Propositional Calculus

$$\frac{\Gamma \vdash B}{\Gamma, A \vdash B}\,[\mathrm{WKN}] \qquad \frac{\Gamma, A, A \vdash B}{\Gamma, A \vdash B}\,[\mathrm{CTR}]$$

Background Work: Substructural Logic

- Two implicit rules in Propositional Calculus

$$\frac{\Gamma \vdash B}{\Gamma, A \vdash B}\,[\mathrm{WKN}] \qquad \frac{\Gamma, A, A \vdash B}{\Gamma, A \vdash B}\,[\mathrm{CTR}]$$

- Control the use of [WKN] and [CTR]

Background Work: Linear Logic

Certain contexts cannot undergo weakening or contraction(**??**)

$$\text{Context} \quad \Gamma, \Delta ::= \epsilon \mid \Gamma, \langle A \rangle \mid \Gamma, [A]$$

$$\begin{aligned} &\Gamma, \langle A \rangle \nvdash \Gamma \\ &\Gamma, \langle A \rangle \nvdash \Gamma, \langle A \rangle, \langle A \rangle \\ &\Gamma, [A] \vdash \Gamma \\ &\Gamma, [A] \vdash \Gamma, [A], [A] \end{aligned}$$

Background Work: Linear Logic

Language

$$\text{Context}\Gamma, \Delta ::= \epsilon \mid \Gamma, \langle A \rangle \mid \Gamma, [A]$$
$$\text{Propostions}A, B, C ::= X \mid !A \mid A \multimap B \mid A \,\&\, B \mid A \otimes B \mid A \oplus B$$

Structural Rules

$$\frac{}{[A] \vdash A}\,[\mathrm{ID}_{[]}] \qquad \frac{}{\langle A \rangle \vdash A}\,[\mathrm{ID}_{\langle\rangle}]$$

$$\frac{\Gamma, \Delta \vdash A}{\Delta, \Gamma \vdash A}\,[\mathrm{EXCH}] \qquad \frac{\Gamma, [A], [A] \vdash B}{\Gamma, [A] \vdash B}\,[\mathrm{CTRN}] \qquad \frac{\Gamma \vdash B}{\Gamma, [A] \vdash B}\,[\mathrm{WKN}]$$

$$\frac{[\Gamma] \vdash A}{[\Gamma] \vdash !A}\,[!\mathrm{I}] \qquad \frac{\Gamma \vdash !A \qquad \Delta, [A] \vdash B}{\Gamma, \Delta \vdash B}\,[!\mathrm{E}]$$

Connective Rules

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}\,[\multimap I] \qquad \frac{\Gamma \vdash A \multimap B \qquad \Delta \vdash A}{\Gamma, \Delta \vdash B}\,[\multimap E]$$

Background Work: Linear Logic

Connective Rules

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}\ [\multimap I] \qquad \frac{\Gamma \vdash A \multimap B \qquad \Delta \vdash A}{\Gamma, \Delta \vdash B}\ [\multimap E]$$

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \,\&\, B}\ [\& I] \qquad \frac{\Gamma \vdash A \,\&\, B}{\Gamma \vdash A}\ [\& E_1] \qquad \frac{\Gamma \vdash A \,\&\, B}{\Gamma \vdash B}\ [\& E_2]$$

Background Work: Linear Logic

Connective Rules

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}\,[\multimap I] \qquad \frac{\Gamma \vdash A \multimap B \qquad \Delta \vdash A}{\Gamma, \Delta \vdash B}\,[\multimap E]$$

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \,\&\, B}\,[\& I] \qquad \frac{\Gamma \vdash A \,\&\, B}{\Gamma \vdash A}\,[\& E_1] \qquad \frac{\Gamma \vdash A \,\&\, B}{\Gamma \vdash B}\,[\& E_2]$$

$$\frac{\Gamma \vdash A \qquad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}\,[\otimes I] \qquad \frac{\Gamma \vdash A \otimes B \qquad \Delta, A, B \vdash C}{\Gamma, \Delta \vdash C}\,[\otimes E]$$

Connective Rules

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}\,[\multimap I] \qquad \frac{\Gamma \vdash A \multimap B \qquad \Delta \vdash A}{\Gamma, \Delta \vdash B}\,[\multimap E]$$

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \,\&\, B}\,[\& I] \qquad \frac{\Gamma \vdash A \,\&\, B}{\Gamma \vdash A}\,[\& E_1] \qquad \frac{\Gamma \vdash A \,\&\, B}{\Gamma \vdash B}\,[\& E_2]$$

$$\frac{\Gamma \vdash A \qquad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}\,[\otimes I] \qquad \frac{\Gamma \vdash A \otimes B \qquad \Delta, A, B \vdash C}{\Gamma, \Delta \vdash C}\,[\otimes E]$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \oplus B}\,[\oplus I_1] \qquad \frac{\Delta \vdash B}{\Delta \vdash A \oplus B}\,[\oplus I_2]$$

$$\frac{\Gamma \vdash A \oplus B \qquad \Delta, A \vdash C \qquad \Delta, B \vdash C}{\Gamma, \Delta \vdash C}\,[\oplus E]$$

Background Work: Linear Logic

- Propositions act like Resources

$$
\begin{array}{rl}
\text{"A consumed to give B"} & A, A \multimap B \vdash B \\
\text{"Both A and B"} & A \otimes B \\
\text{"Choice between A and B"} & A \,\&\, B \\
\text{Cannot create new copies} & A \nvdash A \otimes A \\
\text{Fall back to intuitionstic logic} & !A \multimap B \equiv A \supset B
\end{array}
$$

Background Work: Linear Logic

- Propositions act like Resources

$$
\begin{array}{rl}
\text{“A consumed to give B”} & A, A \multimap B \vdash B \\
\text{“Both A and B”} & A \otimes B \\
\text{“Choice between A and B”} & A \,\&\, B \\
\text{Cannot create new copies} & A \nvdash A \otimes A \\
\text{Fall back to intuistionstic logic} & !A \multimap B \equiv A \supset B
\end{array}
$$

- Curry-Howard Corresondence: Type systems based on Linear Logic
- Active Area of research: L^3(?), F°(?), Linear Haskell(?), Quill(?)
- Resources are first class values

Background Work: Linear Logic

Downside: Asymmetry between implication and conjunction
Multiplicative Fragment:

$$A \otimes B \vdash C \text{ iff } A \vdash B \multimap C$$

Additive Fragment:

$$!A \,\&\, !B \vdash C \text{ iff } !A \vdash !B \multimap C$$

Modality is necessary to avoid over-restriction

Background Work: Qualified Types

General framework to incorporate predicates into type language

$$P \mid \Gamma \vdash M : \sigma$$

"Type of M is σ when predicates in P are satisfied and free variables in M are specified in Γ"?

Background Work: Typing Rules with Qualified Types

HM system with qualified Types:

$$\frac{x : \sigma \in \Gamma}{P \mid \Gamma \vdash x : \sigma}\ [\text{VAR}] \qquad \frac{P \mid \Gamma \vdash M : \sigma \qquad Q \mid \Gamma_x, x : \sigma \vdash N : \tau}{P, Q \mid \Gamma \vdash (\texttt{let}\ x = M\ \texttt{in}\ N) : \tau}\ [\text{LET}]$$

$$\frac{P \mid \Gamma \vdash M : \sigma \qquad t \notin \texttt{fvs}(\Gamma) \cup \texttt{fvs}(P)}{P \mid \Gamma \vdash M : \forall t.\sigma}\ [\forall\ \text{I}] \qquad \frac{P \mid \Gamma \vdash M : \forall t.\sigma}{P \mid \Gamma \vdash M : [\tau/t]\sigma}\ [\forall\ \text{E}]$$

$$\frac{P \mid \Gamma_x, x : \tau \vdash M : \tau'}{P \mid \Gamma \vdash \lambda x.M : \tau \to \tau'}\ [\to \text{I}] \qquad \frac{P \mid \Gamma \vdash M : \tau \to \tau' \qquad P \mid \Gamma \vdash N : \tau}{P \mid \Gamma \vdash MN : \tau'}\ [\to \text{E}]$$

$$\frac{P, \pi \mid \Gamma \vdash M : \rho}{P \mid \Gamma \vdash M : \pi \Rightarrow \rho}\ [\Rightarrow\text{I}] \qquad \frac{P \mid \Gamma \vdash M : \pi \Rightarrow \rho \qquad P \Rightarrow \pi}{P \mid \Gamma \vdash M : \rho}\ [\Rightarrow\text{E}]$$

$\pi \Rightarrow \rho$: π qualifies ρ

$P \Rightarrow \pi$: P entails π

Background Work: Quill

Quill(?): Language with qualified types for linear types and first class polymorphism

Three Predicates:

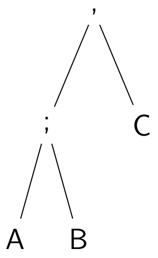
- $\texttt{Un}\ \tau$ If τ does not have resources or can be copied or dropped easily.
- $\texttt{Fun}\ \tau$ If τ is a function type
- $\tau \geq \tau'$ If τ less restricting than τ'

Background Work: Quill

- Unrestricted Types: `Un Int`, `Un` Bool
- Restricted or Linear Types: FileHandle
- Function Types: $\texttt{Fun}\ (\texttt{Int} \rightarrow \texttt{Int})$, $\texttt{Fun}\ (\texttt{String} \rightarrow \texttt{String})$

Background Work: Logic of Bunched Implications (***BI***)

- Contexts as trees, called bunches(**?**)
- Two connective used to combine contexts: $A; B$ or A, B



Background Work: Logic of ***BI***

- Context connective guides choice of implication

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \mathrel{-\!\!*} B}\ [\mathrel{-\!\!*}\mathrm{I}] \qquad\qquad \frac{\Gamma; A \vdash B}{\Gamma \vdash A \twoheadrightarrow B}\ [\twoheadrightarrow\mathrm{I}]$$

- Contraction

$$\begin{array}{ll} A; B \vdash A & A; B \vdash B \\ A, B \nvdash A & A, B \nvdash B \end{array}$$

- Weakening

$$\begin{array}{ll} A; B \vdash A; B; B & A; B \vdash A; A; B \\ A, B \nvdash A, B, B & A, B \nvdash A, A, B \end{array}$$

Background Work: Logic of *BI*

Background Work: $\alpha\lambda$-calculus

Curry-Howard interpretation of logic of ***BI***(?)

$\boxed{\text{Language}}$

$$\begin{array}{rl} \text{Context} & \Gamma, \Delta ::= \{\}_m \mid \{\}_a \mid x : \tau \mid \Gamma, \Delta \mid \Gamma; \Delta \\ \text{Types} & \tau, \upsilon ::= t \mid \iota \mid \tau \twoheadrightarrow \tau \mid \tau \mathrel{-\!\!*} \tau \\ \text{Expressions} & M, N ::= x \mid \lambda x.M \mid \alpha x.M \mid MN \end{array}$$

$\boxed{\text{Typing rules}}$

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau}\,[\text{VAR}] \qquad \frac{\Gamma; \Gamma \vdash M : \tau}{\Gamma \vdash M : \tau}\,[\text{CTR}] \qquad \frac{\Gamma \vdash M : \tau}{\Gamma; \Delta \vdash M : \tau}\,[\text{WKN}]$$

$$\frac{\Gamma_x, x : \tau \vdash M : \tau'}{\Gamma \vdash \lambda x.M : \tau \mathrel{-\!\!*} \tau'}\,[\mathrel{-\!\!*}\text{I}] \qquad \frac{\Gamma \vdash M : \tau \mathrel{-\!\!*} \tau' \qquad \Delta \vdash N : \tau}{\Gamma, \Delta \vdash MN : \tau'}\,[\mathrel{-\!\!*}\text{E}]$$

$$\frac{\Gamma_x; x : \tau \vdash M : \tau'}{\Gamma \vdash \alpha x.M : \tau \twoheadrightarrow \tau'}\,[\twoheadrightarrow\text{I}] \qquad \frac{\Gamma \vdash M : \tau \twoheadrightarrow \tau' \qquad \Delta \vdash N : \tau}{\Gamma; \Delta \vdash MN : \tau'}\,[\twoheadrightarrow\text{E}]$$

QuB

- Quill: Qualified types for type system based on linear logic
- QuB: Qualified types for type system based on logic of ***BI***

QuB: Types and Predicates

$$
\begin{array}{rrl}
\text{Types} & \tau, \upsilon, \phi & ::= t \mid \iota \mid \tau \to \tau \\
& \text{where} & \to \in \{ \overset{!}{\to\!\!\ast}, \to\!\!\ast, \overset{!}{\twoheadrightarrow}, \twoheadrightarrow \} \\
\text{Predicates} & \pi, \omega & ::= \texttt{Un}\ \tau \mid \texttt{SeFun}\ \tau \mid \texttt{ShFun}\ \tau \mid \tau \geq \tau' \\
\text{Qualified Types} & \rho & ::= \tau \mid \pi \Rightarrow \rho \\
\text{Type schemes} & \sigma & ::= \rho \mid \forall t.\sigma
\end{array}
$$

- $\texttt{SeFun}\ \tau$: τ is a function that is separate from its argument
- $\texttt{ShFun}\ \tau$: τ is a function that is in sharing with its argument
- $\texttt{Un}\ \tau$: τ does not have resources or they can be copied/dropped easily

QuB: Types and Predicates

$$
\begin{array}{rrl}
\text{Types} & \tau, \upsilon, \phi & ::= t \mid \iota \mid \tau \to \tau \\
 & \text{where} & \to \in \{ \overset{!}{\mathrel{-\!*}}, \mathrel{-\!*}, \overset{!}{\twoheadrightarrow}, \twoheadrightarrow \} \\
\text{Predicates} & \pi, \omega & ::= \texttt{Un}\ \tau \mid \texttt{SeFun}\ \tau \mid \texttt{ShFun}\ \tau \mid \tau \geq \tau' \\
\text{Qualified Types} & \rho & ::= \tau \mid \pi \Rightarrow \rho \\
\text{Type schemes} & \sigma & ::= \rho \mid \forall t . \sigma
\end{array}
$$

- $\mathrel{-\!*}$: Function type that is separate with its argument
- $\twoheadrightarrow$: Function type that is in sharing with its argument
- $\overset{!}{\mathrel{-\!*}}$, $\overset{!}{\twoheadrightarrow}$: unrestriced versions of $\mathrel{-\!*}$ and $\twoheadrightarrow$

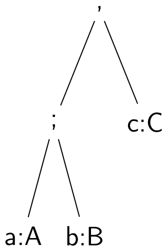
QuB: Language

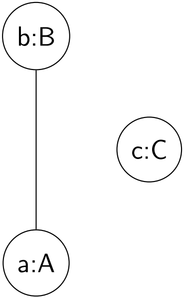
$$\begin{array}{rl} \text{Term Variables} & x, y, z \in \mathrm{Var} \\ \text{Expressions} & M, N ::= x \mid \lambda^{-\!*} x.M \mid \lambda^{\twoheadrightarrow} x.M \mid MN \mid \texttt{let}\ x = M\ \texttt{in}\ N \end{array}$$

- $\lambda^{-\!*} x.M$: Introduces $-\!*$ function
- $\lambda^{\twoheadrightarrow} x.N$: Introduces $\twoheadrightarrow$ function

QuB: Typing Environmnet

- Logic of ***BI***: Contexts are trees
- QuB: Contexts are flattened sharing graphs





- Sharing relation Ψ

$$\forall_{x \in \texttt{dom}(\Psi)}\ x \in \Psi(x) \qquad \text{(reflexive)}$$

$$\forall_{x,y \in \texttt{dom}(\Psi)}\ \text{if } y \in \Psi(x) \text{ then } x \in \Psi(y) \qquad \text{(symmetric)}$$

$$\forall_{x,y,z \in \texttt{dom}(\Psi)}\ \text{if } y \in \Psi(x) \text{ and } z \in \Psi(y)\ \not\Rightarrow z \in \Psi(x) \qquad \text{(non-transitive)}$$

QuB: Typing Environment

"x of type σ is in sharing with $\vec{y}$"

$$(x, \sigma, \vec{y}) \in \Gamma$$

$$\text{Typing Context} \quad \Gamma, \Delta ::= \epsilon \mid \Gamma, x^{\vec{y}} : \sigma$$

$$\begin{aligned}
\texttt{Vars}(\Gamma, x^{\vec{y}} : \tau) &= \texttt{Vars}(\Gamma) \cup \{x\} \\
\texttt{Shared}(\Gamma, x^{\vec{y}} : \tau) &= \texttt{Shared}(\Gamma) \cup \{\vec{y}\} \\
\texttt{Used}(\Gamma) &= \texttt{Vars}(\Gamma) \cup \texttt{Shared}(\Gamma)
\end{aligned}$$

$$(\Gamma, x^{\vec{y}} : \tau)^{[a \mapsto \vec{b}]} = \begin{cases} a \notin \vec{y} & (\Gamma^{[a \mapsto \vec{b}]}, x^{\vec{y}} : \tau) \\ a \in \vec{y} & (\Gamma^{[a \mapsto \vec{b}]}, x^{(\vec{y} \backslash a) \cup \vec{b}} : \tau) \end{cases}$$

$$\Gamma^{[\vec{a} \mapsto \vec{b}]} = (\ldots((\Gamma^{[a_1 \mapsto \vec{b}]})^{[a_2 \mapsto \vec{b}]})^{\cdots})^{[a_n \mapsto \vec{b}]}$$

$$\begin{aligned}
\Gamma \circledast \Gamma' &= \Gamma \sqcup \Gamma' \qquad \texttt{if } \texttt{Vars}(\Gamma) \# \texttt{Used}(\Gamma') \wedge \texttt{Vars}(\Gamma') \# \texttt{Used}(\Gamma) \\
\Gamma \oplus \Gamma' &= \Gamma \sqcup \Gamma' \qquad \texttt{if } \texttt{Used}(\Gamma) = \texttt{Used}(\Gamma')
\end{aligned}$$

QuB: Typing Rules

Structural Rules

$$\frac{}{P \mid x^{\vec{y}} : \sigma \vdash x : \sigma}\ [\text{ID}]$$

$$\frac{P \mid \Gamma \circledast \Delta \circledast \Delta \vdash M : \sigma \qquad P \vdash \Delta\ \mathbf{un}}{P \mid \Gamma \circledast \Delta \vdash M : \sigma}\ [\text{CTR-UN}] \qquad \frac{P \mid \Gamma \oplus \Delta \oplus \Delta \vdash M : \sigma}{P \mid \Gamma \oplus \Delta \vdash M : \sigma}\ [\text{CTR-SH}]$$

$$\frac{P \mid \Gamma \vdash M : \sigma \qquad P \vdash \Delta\ \mathbf{un}}{P \mid \Gamma \circledast \Delta \vdash M : \sigma}\ [\text{WKN-UN}] \qquad \frac{P \mid \Gamma \vdash M : \sigma}{P \mid \Gamma \oplus \Delta \vdash M : \sigma}\ [\text{WKN-SH}]$$

QuB: Typing Rules

Connective Rules

$$\frac{P \mid \Gamma \vdash M : \sigma \qquad P' \mid \Gamma'_x, x : \sigma \vdash N : \tau}{P \cup P' \mid \Gamma \sqcup \Gamma' \vdash (\texttt{let}\ x = M\ \texttt{in}\ N) : \tau}\ [\text{LET}]$$

$$\frac{P \mid \Gamma \vdash M : \sigma \qquad t \notin \texttt{fvs}(\Gamma) \cup \texttt{fvs}(P)}{P \mid \Gamma \vdash M : \forall t.\sigma}\ [\forall\ \text{I}] \qquad \frac{P \mid \Gamma \vdash M : \forall t.\sigma}{P \mid \Gamma \vdash M : [\tau \backslash t]\sigma}\ [\forall\ \text{E}]$$

$$\frac{P, \pi \mid \Gamma \vdash M : \rho}{P \mid \Gamma \vdash M : \pi \Rightarrow \rho}\ [\Rightarrow\ \text{I}] \qquad \frac{P \mid \Gamma \vdash M : \pi \Rightarrow \rho \qquad P \vdash \pi}{P \mid \Gamma \vdash M : \rho}\ [\Rightarrow\ \text{E}]$$

$$\frac{\begin{array}{c} P \Rightarrow \texttt{ShFun}\ \phi \qquad P \vdash \Gamma \geq \phi \\ P \mid \Gamma^{[\varnothing \mapsto \{x\}]}, x^{\mathsf{Vars}(\Gamma)} : \tau \vdash M : \tau' \end{array}}{P \mid \Gamma \vdash \lambda^{\twoheadrightarrow} x.M : \phi\tau\tau'}\ [\twoheadrightarrow\ \text{I}] \qquad \frac{\begin{array}{c} P \Rightarrow \texttt{ShFun}\ \phi \\ P \mid \Gamma \vdash M : \phi\tau\tau' \qquad P \mid \Gamma' \vdash N : \tau \end{array}}{P \mid \Gamma \oplus \Gamma' \vdash MN : \tau'}\ [\twoheadrightarrow\ \text{E}]$$

$$\frac{\begin{array}{c} P \Rightarrow \texttt{SeFun}\ \phi \qquad P \vdash \Gamma \geq \phi \\ P \mid \Gamma, x^{\varnothing} : \tau \vdash M : \tau' \end{array}}{P \mid \Gamma \vdash \lambda^{-\!\!*} x.M : \phi\tau\tau'}\ [-\!\!*\ \text{I}] \qquad \frac{\begin{array}{c} P \Rightarrow \texttt{SeFun}\ \phi \\ P \mid \Gamma \vdash M : \phi\tau\tau' \qquad P \mid \Gamma' \vdash N : \tau \end{array}}{P \mid \Gamma \circledast \Gamma' \vdash MN : \tau'}\ [-\!\!*\ \text{E}]$$

QuB: Typing Terms

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{}{\varnothing \mid y^{\varnothing} : \tau' \vdash y : \tau'}\,[\mathrm{ID}]
\qquad
\dfrac{
\dfrac{}{\varnothing \mid x^{\varnothing} : \tau \vdash x : \tau}\,[\mathrm{ID}]
\quad
\dfrac{}{\varnothing \mid f^{\varnothing} : \tau \mathrel{-\!\!*} \tau' \mathrel{-\!\!*} \upsilon \vdash f : \tau \mathrel{-\!\!*} \tau' \mathrel{-\!\!*} \upsilon}\,[\mathrm{ID}]
}{\varnothing \mid x^{\varnothing} : \tau \circledast f^{\varnothing} : \tau \mathrel{-\!\!*} \tau' \mathrel{-\!\!*} \upsilon \vdash fx : (\tau' \mathrel{-\!\!*} \upsilon)}\,[\mathrel{-\!\!*}\mathrm{E}]
}{\varnothing \mid y^{\varnothing} : \tau' \circledast x^{\varnothing} : \tau \circledast f^{\varnothing} : \tau \mathrel{-\!\!*} \tau' \mathrel{-\!\!*} \upsilon \vdash fxy : \upsilon}\,[\mathrel{-\!\!*}\mathrm{E}]
}{\varnothing \mid x^{\varnothing} : \tau \circledast y^{\varnothing} : \tau' \circledast f^{\varnothing} : \tau \mathrel{-\!\!*} \tau' \mathrel{-\!\!*} \upsilon \vdash fxy : \upsilon}\,[\mathit{EXCH}]
}{\varnothing \mid x^{\varnothing} : \tau \circledast y^{\varnothing} : \tau' \vdash \lambda^{\mathrel{-\!\!*}} f.fxy : (\tau \mathrel{-\!\!*} \tau' \mathrel{-\!\!*} \upsilon) \mathrel{-\!\!*} \upsilon}\,[\mathrel{-\!\!*}\ \mathrm{I}]
}{\varnothing \mid x^{\varnothing} : \tau \vdash \lambda^{\mathrel{-\!\!*}} y.\lambda^{\mathrel{-\!\!*}} f.fxy : \tau' \mathrel{-\!\!*} (\tau \mathrel{-\!\!*} \tau' \mathrel{-\!\!*} \upsilon) \mathrel{-\!\!*} \upsilon}\,[\mathrel{-\!\!*}\ \mathrm{I}]
}{\varnothing \mid I \vdash \lambda^{\mathrel{-\!\!*}} x.\lambda^{\mathrel{-\!\!*}} y.\lambda^{\mathrel{-\!\!*}} f.fxy : \tau \mathrel{-\!\!*} \tau' \mathrel{-\!\!*} (\tau \mathrel{-\!\!*} \tau' \mathrel{-\!\!*} \upsilon) \mathrel{-\!\!*} \upsilon}\,[\equiv]
$$

QuB: Typing Terms

$$
\cfrac{
  \cfrac{
    \cfrac{}{\varnothing \mid y^{\boldsymbol{xf}} : \tau' \vdash y : \tau'}\,[\mathrm{ID}]
    \qquad
    \cfrac{
      \cfrac{}{\varnothing \mid x^{\boldsymbol{fy}} : \tau \vdash x : \tau}\,[\mathrm{ID}]
      \qquad
      \cfrac{}{\varnothing \mid f^{\boldsymbol{xy}} : \tau \twoheadrightarrow \tau' \twoheadrightarrow \upsilon \vdash f : \tau \twoheadrightarrow \tau' \twoheadrightarrow \upsilon}\,[\mathrm{ID}]
    }{\varnothing \mid x^{\boldsymbol{xy}} : \tau \oplus f^{\boldsymbol{xy}} : \tau \twoheadrightarrow \tau' \twoheadrightarrow \upsilon \vdash fx : (\tau' \twoheadrightarrow \upsilon)}\,[\mathrel{-\!\!\ast}\ E]
  }{
    \cfrac{
      \cfrac{
        \cfrac{\varnothing \mid y^{\boldsymbol{xf}} : \tau' \oplus x^{\boldsymbol{yf}} : \tau \oplus f^{\boldsymbol{xy}} : \tau \twoheadrightarrow \tau' \twoheadrightarrow \upsilon \vdash fxy : \upsilon}{\varnothing \mid x^{\boldsymbol{yf}} : \tau \oplus y^{\boldsymbol{xf}} : \tau' \oplus f^{\boldsymbol{xy}} : \tau \twoheadrightarrow \tau' \twoheadrightarrow \upsilon \vdash fxy : \upsilon}\,[EXCH]
      }{\varnothing \mid x^{\boldsymbol{y}} : \tau \oplus y^{\boldsymbol{x}} : \tau' \vdash \lambda^{\twoheadrightarrow} f.fxy : (\tau \twoheadrightarrow \tau' \twoheadrightarrow \upsilon) \twoheadrightarrow \upsilon}\,[\twoheadrightarrow I]
    }{\varnothing \mid x^{\varnothing} : \tau \vdash \lambda^{\twoheadrightarrow} y.\lambda^{\twoheadrightarrow} f.fxy : \tau' \twoheadrightarrow (\tau \twoheadrightarrow \tau' \twoheadrightarrow \upsilon) \twoheadrightarrow \upsilon}\,[\twoheadrightarrow I]
  }\,[\twoheadrightarrow E]
}{\varnothing \mid \boldsymbol{I} \vdash \lambda^{\mathrel{-\!\!\ast}} x.\lambda^{\twoheadrightarrow} y.\lambda^{\twoheadrightarrow} f.fxy : \tau \mathrel{-\!\!\ast} \tau' \twoheadrightarrow (\tau \twoheadrightarrow \tau' \twoheadrightarrow \upsilon) \twoheadrightarrow \upsilon}\,[\mathrel{-\!\!\ast}\ \mathrm{I}]
$$

QuB: Syntax Directed Typing Rules

Term structure $\leftrightarrow$ Typing Rule

QuB: Syntax Directed Typing Rules

$$\frac{P \vdash \Gamma_{\vec{y}}\ \mathtt{un} \qquad (P \Rightarrow \tau) \sqsubseteq \sigma}{P \mid \Gamma, x^{\vec{y}} : \sigma \vdash^{s} x : \tau}\ [\mathrm{VAR}^{s}]$$

$$\frac{\begin{array}{c} Q \mid (\Gamma'_x \oplus \Gamma''_x) \circledast \Delta \vdash^{s} M : \upsilon \qquad P \vdash \Delta\ \mathtt{un} \\ P \mid (\Gamma_x, x^{\varnothing} : \sigma) \oplus \Gamma''_x \circledast \Delta \vdash^{s} N : \tau \qquad \sigma = \mathtt{Gen}(\{\Gamma' \oplus \Gamma''_x \circledast \Delta\}, Q \Rightarrow \upsilon) \end{array}}{P, Q \mid (\Gamma \circledast \Gamma') \oplus \Gamma'' \circledast \Delta \vdash^{s} (\mathtt{let}\ x = M\ \mathtt{in}\ N) : \tau}\ [\mathrm{Let}^{s}]$$

$$[\mathrm{VAR}^{s}] \equiv [\mathrm{ID}] + [\forall\mathrm{E}] + [\Rightarrow\!\mathrm{E}]$$

$$[\mathrm{LET}^{s}] \equiv [\mathrm{LET}] + [\forall\mathrm{I}] + [\Rightarrow\!\mathrm{I}]$$

$$\frac{P \Rightarrow \mathtt{SeFun}\ \phi \qquad P \vdash \Gamma \geq \phi \qquad P \mid \Gamma \circledast x^{\varnothing} : \tau \vdash^{s} M : \upsilon}{P \mid \Gamma \vdash^{s} \lambda^{\multimap} x.M : \phi\tau\upsilon}\,[\multimap \mathrm{I}^{s}] \qquad \frac{P \Rightarrow \mathtt{ShFun}\ \phi \qquad P \vdash \Gamma \geq \phi \qquad P \mid \Gamma^{[\varnothing \mapsto \{x\}]} \oplus x^{\mathtt{Vars}(\Gamma)} : \tau \vdash^{s} M : \upsilon}{P \mid \Gamma \vdash^{s} \lambda^{\twoheadrightarrow} x.M : \phi\tau\upsilon}\,[\twoheadrightarrow \mathrm{I}^{s}]$$

$$\frac{P \mid \Gamma \circledast \Delta \vdash^{s} M : \phi\upsilon\tau \qquad P \mid \Gamma' \circledast \Delta \vdash^{s} N : \upsilon \qquad P \vdash \Delta\ \mathtt{un} \qquad (\Gamma \mathbin{\tilde{\oplus}} \Gamma' \wedge (P \Rightarrow \mathtt{ShFun}\ \phi)) \vee (\Gamma \mathbin{\tilde{\circledast}} \Gamma' \wedge (P \Rightarrow \mathtt{SeFun}\ \phi))}{P \mid \Gamma \sqcup \Gamma' \circledast \Delta \vdash^{s} MN : \tau}\,[\mathrm{App}^{s}]$$

Context Operations

$$\Gamma \mathbin{\tilde{\circledast}} \Delta = \mathtt{Used}(\Gamma) \# \mathtt{Vars}(\Delta) \wedge \mathtt{Used}(\Delta) \# \mathtt{Vars}(\Gamma)$$

$$\Gamma \mathbin{\tilde{\oplus}} \Delta = \mathtt{Used}(\Gamma) = \mathtt{Used}(\Delta)$$

$$[\twoheadrightarrow \mathrm{I}] \equiv [\twoheadrightarrow \mathrm{I}] \qquad [\multimap \mathrm{I}] \equiv [\multimap \mathrm{I}]$$

$$[\mathrm{APP}^{s}] \equiv [\twoheadrightarrow \mathrm{E}] + [\multimap \mathrm{E}]$$

QuB: Algorithm $\mathcal{W}$

QuB

Multiplicative Products

Additive Products

Sums

Multilicative Implication

Additive Implication

Conclusion and Future Work

What next?

Conclusion and Future Work

Thank You!

Conclusion and Future Work

Q & A