Unifying Function- and Argument-First Bidirectional Type Systems

吉岡拓真*1　関山太朗*2　五十嵐淳*1
*1：京都大学　*2：国立情報学研究所

双方向性型付け [Pierce and Turner, 1998] は**型合成**と**型検査**を使い分ける手法

型合成 $\Gamma \vdash e \Rightarrow A$ 　型を出力

型検査 $\Gamma \vdash e \Leftarrow A$ 　型は入力

研究目的：関数先行と引数先行の双方向性型付けの統一

関数の型を先に合成 [Dunfield and Krishnaswami, 2013]

合成した引数型を再利用して型検査

$$\frac{\Gamma \vdash e_1 \Rightarrow A \to B \qquad \Gamma \vdash e_2 \Leftarrow A}{\Gamma \vdash e_1\ e_2 \Rightarrow B}$$

引数の型を先に合成 [Xie and Oliveira, 2018]

合成した引数型を入力　返り値型を出力

$$\frac{\Gamma \mid \emptyset \vdash e_2 \Rightarrow A \qquad \Gamma \mid \Psi, A \vdash e_1 \Rightarrow A \to B}{\Gamma \mid \Psi \vdash e_1\ e_2 \Rightarrow B}$$

動機：関数先行と引数先行は型付け能力が非互換

$(\lambda(f : \underline{(\forall a.a \to a) \to (\mathsf{int} \times \mathsf{bool})}).f)\ (\lambda g.(g\ 42, g\ \mathsf{true}))$ 　✓関数先行　✗引数先行

$(\lambda f.f)\ (\lambda(\underline{g : \forall a.a \to a}).(g\ 42, g\ \mathsf{true}))$ 　✗関数先行　✓引数先行

アプローチ：Boxy Type [Vytiniotis et al., 2006] による型付け規則の記述

型の中で**合成**するべき部分をBoxで囲って明示する記法

Types $A, B ::= \cdots \mid A \to B$

Boxy Types $P, Q ::= \cdots \mid P \to Q \mid \boxed{A}$

関数先行の規則

関数の引数型 A を合成

$$\frac{\Gamma \vdash e_1 : \boxed{A} \to P \qquad \Gamma \vdash e_2 : A}{\Gamma \vdash e_1 \looparrowright e_2 : P}$$

引数先行の規則

Boxの外側は型検査に利用

引数の型 A を合成

$$\frac{\Gamma \vdash e_1 : A \to P \qquad \Gamma \vdash e_2 : \boxed{A}}{\Gamma \vdash e_1 \looparrowleft e_2 : P}$$

関数先行を用いるという注釈　表面言語から挿入　引数先行を用いるという注釈

貢献

- 関数先行・引数先行を統一した型システム
  - 既存の体系 [Dunfield and Krishnaswami, 2013] [Xie and Oliveira, 2018] を包含することをAbellaで証明
- Worklist [Zhao et al., 2019] に基づく型付けアルゴリズムの提案
  - 我々の型システムに対して sound かつ、既存の体系に対して complete

今後の課題

- Let 多相や Explicit Type Application への拡張
- Level-Based Type Inference [Fan et al., 2025] への適応