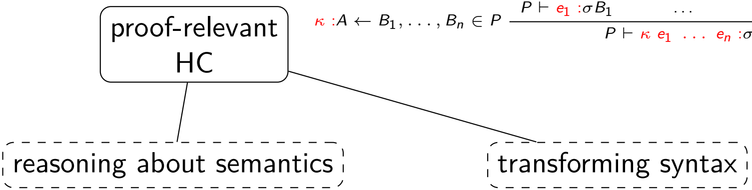
Proof-Relevant Resolution for Elaboration of Programming Languages

František Farka

Heriot-Watt University, and
University of St Andrews

ICLP DC, Oxford, July 17, 2018

Proof-Relevant Resolution for Elaboration of Programming Languages



$$\kappa : A \leftarrow B_1, \ldots, B_n \in P \quad \frac{P \vdash e_1 :_\sigma B_1 \quad \ldots \quad P \vdash e_n :_\sigma B_n}{P \vdash \kappa\ e_1\ \ldots\ e_n :_\sigma A}$$

```
data OddList a    = OCons a (EvenList a)
data EvenList a   = Nil | ECons a (OddList a)

instance Eq a, Eq (OddList a) ⇒ Eq (EvenList a)
  eq (OCons a as) (OCons b bs) = eq a b && eq as bs
instance Eq a, Eq (EvenList a)
  ...
```

```
data maybe_A (a : A)   : Bool → type  where
     nothing           : maybe_A ff
     just              : A → maybe_A tt

fromJust : maybe_A tt → A
fromJust (just x) = x
```

Motivation

- 90's - Haskell type classes introduced (Wadler and Blott; 1989), (Hall *et al.*; 1996)
- Attracted attention of ATP community for its connection with LP
- 2003 - Lämmel *et al.*- extension to non-terminating resolution (Lämmel and Peyton Jones; 2005)
- 2016 - Fu *et al.* study of properties of type class resolution (TCR) from type-theoretic perspective (Fu and Komendantskaya; 2016), (Fu *et al.*; 2016)

Problem by an example

Example (Equality on `Pairs`)

```
data Pair a b = Pair a b

class Eq a where
  eq :: a → a → Bool

instance (Eq a, Eq b) ⇒ Eq (Pair a b) where
  eq (Pair x1 y1) (Pair x2 y2) = eq x1 x2 && eq y1 y2
instance Eq Int where
  eq x y = primtiveIntEq x y

member :: Eq a ⇒ a → List a → Bool
member = ...
```

Example (Φ_{Pair})

$$
\begin{aligned}
\kappa_1 : \mathrm{eq}(x),\ \mathrm{eq}(y) &\Rightarrow \mathrm{eq}(\mathtt{pair}(x, y)) \\
\kappa_2 : &\Rightarrow \mathrm{eq}(\mathtt{int})
\end{aligned}
$$

Problem by an example

Example (Equality on `Pairs`)

```
data Pair a b = Pair a b

class Eq a where
  eq :: a → a → Bool

instance (Eq a, Eq b) ⇒ Eq (Pair a b) where
  eq (Pair x1 y1) (Pair x2 y2) = eq x1 x2 && eq y1 y2
instance Eq Int where
  eq x y = primtiveIntEq x y

member :: Eq a ⇒ a → List a → Bool
member = ...
```

Example (Φ_{Pair})

$$\begin{aligned} \kappa_1 : \mathrm{eq}(x),\ \mathrm{eq}(y) &\Rightarrow \mathrm{eq}(\mathtt{pair}(x, y)) \\ \kappa_2 : \quad &\Rightarrow \mathrm{eq}(\mathtt{int}) \end{aligned}$$

$$\mathrm{eq}(\mathtt{pair}(\mathtt{int}, \mathtt{int})) \to_{\kappa_1} \mathrm{eq}(\mathtt{int}), \mathrm{eq}(\mathtt{int}) \to_{\kappa_2} \mathrm{eq}(\mathtt{int}) \to_{\kappa_2} \emptyset$$

Type class resolution

Definition (Type class resolution)

$$\text{if } (e : B_1, \ldots, B_n \Rightarrow A) \in \Phi \; \frac{\Phi \vdash e_1 : \sigma B_1 \quad \cdots \quad \Phi \vdash e_n : \sigma B_n}{\Phi \vdash e \; e_1 \cdots e_n : \sigma A} \quad (\textsc{Lp-m})$$

$$\text{if HNF}(e) \; \frac{\Phi, (\alpha : B_1, \ldots, B_n \Rightarrow A) \vdash e : B_1, \ldots, B_n \Rightarrow A}{\Phi \vdash \nu\alpha.e : B_1, \ldots, B_n \Rightarrow A} \quad (\textsc{Nu})$$

$$\frac{\Phi, (\beta_1 : \Rightarrow B_1), \ldots, (\beta_n : \Rightarrow B_n) \vdash e : A}{\Phi \vdash \lambda\beta_1, \ldots, \beta_n.e : B_1, \ldots, B_n \Rightarrow A} \quad (\textsc{Lam})$$

Type Class Resolution - by LP-M

Example

$$\begin{aligned} \kappa_1 &: \mathtt{eq}(x),\ \mathtt{eq}(y) \Rightarrow \mathtt{eq}(\mathtt{pair}(x, y)) \\ \kappa_2 &: \Rightarrow \mathtt{eq}(\mathtt{int}) \end{aligned}$$

$$\dfrac{\dfrac{}{\Phi_{Pair} \vdash \kappa_2 : \mathtt{eq}(\mathtt{int})}\ \textsc{Lp-m} \qquad \dfrac{}{\Phi_{Pair} \vdash \kappa_2 : \mathtt{eq}(\mathtt{int})}\ \textsc{Lp-m}}{\Phi_{Pair} \vdash \kappa_1\kappa_2\kappa_2 : \mathtt{eq}(\mathtt{pair}(\mathtt{int}, \mathtt{int}))}\ \textsc{Lp-m}$$

Type Class Resolution - by Lp-m

Example

$$
\begin{aligned}
\kappa_1 &: \mathtt{eq}(x),\ \mathtt{eq}(y) &\Rightarrow \mathtt{eq}(\mathtt{pair}(x,y)) \\
\kappa_2 &: &\Rightarrow \mathtt{eq}(\mathtt{int})
\end{aligned}
$$

$$
\dfrac{\dfrac{}{\Phi_{Pair} \vdash \kappa_2 : \mathtt{eq}(\mathtt{int})}\ \textsc{Lp-m} \qquad \dfrac{}{\Phi_{Pair} \vdash \kappa_2 : \mathtt{eq}(\mathtt{int})}\ \textsc{Lp-m}}{\Phi_{Pair} \vdash \kappa_1\kappa_2\kappa_2 : \mathtt{eq}(\mathtt{pair}(\mathtt{int},\mathtt{int}))}\ \textsc{Lp-m}
$$

Theorem

Let Φ be an axiom environment for a logic program P, and let $\Phi \vdash e : A$ by the Lp-m rule. Then $P \vDash_{ind} A$.

Data types *EvenList* and *OddList*

```
data OddList a   =  OCons a (EvenList a)
data EvenList a  =  Nil | ECons a (OddList a)

instance Eq a, Eq (OddList a) ⇒ Eq (EvenList a)
    eq (OCons a as) (OCons b bs)    = eq a b && eq as bs

instance Eq a, Eq (EvenList a) ⇒ Eq (OddList a)
    ...
```

Example ($\Phi_{EvenOdd}$)

$$
\begin{aligned}
\kappa_1 &: \mathrm{eq}(x),\ \mathrm{eq}(\mathtt{evenList}(x)) \Rightarrow \mathrm{eq}(\mathtt{oddList}(x))\\
\kappa_2 &: \mathrm{eq}(x),\ \mathrm{eq}(\mathtt{oddList}(x)) \Rightarrow \mathrm{eq}(\mathtt{evenList}(x))\\
\kappa_3 &: \Rightarrow \mathrm{eq}(\mathtt{int})
\end{aligned}
$$

Data types *EvenList* and *OddList*

```
data OddList a   =  OCons a (EvenList a)
data EvenList a  =  Nil | ECons a (OddList a)

instance Eq a, Eq (OddList a) ⇒ Eq (EvenList a)
    eq (OCons a as) (OCons b bs)    = eq a b && eq as bs

instance Eq a, Eq (EvenList a) ⇒ Eq (OddList a)
    ...
```

Example ($\Phi_{EvenOdd}$)

$$\begin{aligned}
\kappa_1 &: \mathtt{eq}(x),\ \mathtt{eq}(\mathtt{evenList}(x)) && \Rightarrow \mathtt{eq}(\mathtt{oddList}(x)) \\
\kappa_2 &: \mathtt{eq}(x),\ \mathtt{eq}(\mathtt{oddList}(x)) && \Rightarrow \mathtt{eq}(\mathtt{evenList}(x)) \\
\kappa_3 &: && \Rightarrow \mathtt{eq}(\mathtt{int})
\end{aligned}$$

$$\underline{\mathtt{eq}(\mathtt{evenList}(\mathtt{int}))} \to_{\kappa_2} \mathtt{eq}(\mathtt{int}),\ \mathtt{eq}(\mathtt{oddList}(\mathtt{int})) \to_{\kappa_3}$$

$$\mathtt{eq}(\mathtt{oddList}(\mathtt{int})) \to_{\kappa_1} \mathtt{eq}(\mathtt{int}),\ \mathtt{eq}(\mathtt{evenList}(\mathtt{int})) \to_{\kappa_3}$$

$$\underline{\mathtt{eq}(\mathtt{evenList}(\mathtt{int}))} \to_{\kappa_2} \ldots$$

Type Class Resolution - by LP-M and NU

$\Phi_{EvenOdd}$

$$
\begin{array}{lll}
\kappa_1 : & \mathtt{eq}(x),\ \mathtt{eq}(\mathtt{evenList}(x)) & \Rightarrow \mathtt{eq}(\mathtt{oddList}(x)) \\
\kappa_2 : & \mathtt{eq}(x),\ \mathtt{eq}(\mathtt{oddList}(x)) & \Rightarrow \mathtt{eq}(\mathtt{evenList}(x)) \\
\kappa_3 : & & \Rightarrow \mathtt{eq}(\mathtt{int})
\end{array}
$$

Example

$$
\dfrac{
\dfrac{
\dfrac{}{\begin{array}{c}\kappa_3 : \mathtt{eq}(\mathtt{int}) \\ \vdash \kappa_3 : \mathtt{eq}(\mathtt{int})\end{array}}
\qquad
\dfrac{
\dfrac{}{\begin{array}{c}\kappa_3 : \mathtt{eq}(\mathtt{int}) \\ \vdash \kappa_3 : \mathtt{eq}(\mathtt{int})\end{array}}
\quad
\dfrac{}{\begin{array}{c}\alpha : \Rightarrow \mathtt{eq}(\mathtt{evenList}(\mathtt{int})) \\ \vdash \alpha : \mathtt{eq}(\mathtt{evenList}(\mathtt{int}))\end{array}}
}{\Phi_{EvenOdd}, \alpha : \_ \vdash \kappa_1\kappa_3\alpha : \mathtt{eq}(\mathtt{oddList}(\mathtt{int}))}
}{\Phi_{EvenOdd}, \alpha : \_ \vdash \kappa_2\kappa_3(\kappa_1\kappa_3\alpha) : \mathtt{eq}(\mathtt{evenList}(\mathtt{int}))}
}{\Phi_{EvenOdd} \vdash \nu\alpha.\kappa_2\kappa_3(\kappa_1\kappa_3\alpha) : \mathtt{eq}(\mathtt{evenList}(\mathtt{int}))}\ \textsc{Nu}
$$

Type Class Resolution - by LP-M and NU

$\Phi_{EvenOdd}$

$$
\begin{aligned}
\kappa_1 &: \mathtt{eq}(x),\ \mathtt{eq}(\mathtt{evenList}(x)) && \Rightarrow \mathtt{eq}(\mathtt{oddList}(x)) \\
\kappa_2 &: \mathtt{eq}(x),\ \mathtt{eq}(\mathtt{oddList}(x)) && \Rightarrow \mathtt{eq}(\mathtt{evenList}(x)) \\
\kappa_3 &: && \Rightarrow \mathtt{eq}(\mathtt{int})
\end{aligned}
$$

Example

$$
\dfrac{\dfrac{\dfrac{}{\kappa_3 : \mathtt{eq}(\mathtt{int})}}{\vdash \kappa_3 : \mathtt{eq}(\mathtt{int})} \qquad \dfrac{\dfrac{}{\kappa_3 : \mathtt{eq}(\mathtt{int}) \vdash \kappa_3 : \mathtt{eq}(\mathtt{int})} \qquad \dfrac{}{\alpha : \Rightarrow \mathtt{eq}(\mathtt{evenList}(\mathtt{int})) \vdash \alpha : \mathtt{eq}(\mathtt{evenList}(\mathtt{int}))}}{\Phi_{EvenOdd}, \alpha : \_ \vdash \kappa_1\kappa_3\alpha : \mathtt{eq}(\mathtt{oddList}(\mathtt{int}))}}{\dfrac{\Phi_{EvenOdd}, \alpha : \_ \vdash \kappa_2\kappa_3(\kappa_1\kappa_3\alpha) : \mathtt{eq}(\mathtt{evenList}(\mathtt{int}))}{\Phi_{EvenOdd} \vdash \nu\alpha.\kappa_2\kappa_3(\kappa_1\kappa_3\alpha) : \mathtt{eq}(\mathtt{evenList}(\mathtt{int}))}\ \textsc{Nu}}
$$

Theorem

Let Φ be an axiom environment for a logic program P, and let F be an atomic formula. If $\Phi \vdash e : F$ by the LP-M *and* NU *rules, then $P \vDash_{coind} F$.*

Data type *Bush*

```
data Bush a = Nil | Cons a (Bush (Bush a))

instance Eq a, Eq (Bush (Bush a)) ⇒ Eq (Bush a)
    ...
```

Example (Φ_{Bush})

$$\begin{aligned} \kappa_1 &: & &\Rightarrow \mathtt{eq}(\mathtt{int}) \\ \kappa_2 &: \mathtt{eq}(x),\ \mathtt{eq}(\mathtt{bush}(\mathtt{bush}(x))) & &\Rightarrow \mathtt{eq}(\mathtt{bush}(x)) \end{aligned}$$

Data type *Bush*

```
data Bush a = Nil | Cons a (Bush (Bush a))

instance Eq a, Eq (Bush (Bush a)) ⇒ Eq (Bush a)
    ...
```

Example (Φ_{Bush})

$$\begin{aligned} \kappa_1 &: && \Rightarrow \mathtt{eq}(\mathtt{int}) \\ \kappa_2 &: \mathtt{eq}(x),\ \mathtt{eq}(\mathtt{bush}(\mathtt{bush}(x))) && \Rightarrow \mathtt{eq}(\mathtt{bush}(x)) \end{aligned}$$

$$\mathtt{eq}(\mathtt{bush}(\mathtt{int})) \to_{\kappa_2} \mathtt{eq}(\mathtt{int}), \mathtt{eq}(\mathtt{bush}(\mathtt{bush}(\mathtt{int})) \to_{\kappa_1} \ldots \to_{\kappa_2} \\ \mathtt{eq}(\mathtt{bush}(\mathtt{int})), \mathtt{eq}(\mathtt{bush}(\mathtt{bush}(\mathtt{bush}(\mathtt{int}))) \to_{\kappa_1} \ldots$$

Type class resolution - by LP-M, NU, and LAM

Φ_{Bush}

$$\kappa_1 : \quad \Rightarrow \mathtt{eq}(\mathtt{int})$$

$$\kappa_2 : \mathtt{eq}(x),\ \mathtt{eq}(\mathtt{bush}(\mathtt{bush}(x))) \Rightarrow \mathtt{eq}(\mathtt{bush}(x))$$

Example

$$\dfrac{\dfrac{}{\Phi_{Bush} \vdash \kappa_1 : \mathtt{eq}(\mathtt{int})} \qquad \dfrac{\dfrac{\dfrac{\vdots}{\Phi_{Bush}, (\alpha : \mathtt{eq}(x) \Rightarrow \mathtt{eq}(\mathtt{bush}(x))), (\beta : \ \Rightarrow \mathtt{eq}(x)) \vdash \kappa_2\beta(\alpha(\alpha\beta)) : \mathtt{eq}(\mathtt{bush}(x))}}{\Phi_{Bush}, (\alpha : \_) \vdash \lambda\beta.\kappa_2\beta(\alpha(\alpha\beta)) : \mathtt{eq}(x) \Rightarrow \mathtt{eq}(\mathtt{bush}(x))}\ \textsc{Lam}}{\Phi_{Bush} \vdash \nu\alpha.\lambda\beta.\kappa_2\beta(\alpha(\alpha\beta)) : \mathtt{eq}(x) \Rightarrow \mathtt{eq}(\mathtt{bush}(x))}\ \textsc{Nu}}{\Phi_{Bush} \vdash (\nu\alpha.\lambda\beta.\kappa_2\beta(\alpha(\alpha\beta)))\kappa_1 : \mathtt{eq}(\mathtt{bush}(\mathtt{int}))}$$

Soundness of TCR

Theorem

Let Φ be an axiom environment for a logic program P, and let be $\Phi \vdash e : F$ for a formula F by the LP-M, LAM, *and* NU *rules. Then $P \vDash_{coind} F$.*

Soundness of TCR

Theorem

Let Φ be an axiom environment for a logic program P, and let be $\Phi \vdash e : F$ for a formula F by the LP-M, LAM, *and* NU *rules. Then $P \vDash_{coind} F$.*

Example

Consider $\Phi_A = \{\kappa : A \Rightarrow A\}$.
Then $\Phi_A \vdash \nu\alpha.\kappa\alpha : A$ but $P_A \nvDash_{ind} A$.

Soundness of TCR

Theorem

Let Φ be an axiom environment for a logic program P, and let be $\Phi \vdash e : F$ for a formula F by the LP-M, LAM, *and* NU *rules. Then $P \vDash_{coind} F$.*

Example

Consider $\Phi_A = \{\kappa : A \Rightarrow A\}$.
Then $\Phi_A \vdash \nu\alpha.\kappa\alpha : A$ but $P_A \nvDash_{ind} A$.
TCR is not inductively sound

Conclusion

Conclusion

- We provide a uniform analysis of soundness of type class resolution
- We establish coinductive soundness of corecursive type class resolution
- We demonstrate that corecursive type class resolution is not sound inductively

Conclusion

Conclusion

- We provide a uniform analysis of soundness of type class resolution
- We establish coinductive soundness of corecursive type class resolution
- We demonstrate that corecursive type class resolution is not sound inductively

Thank you. Questions?