Properties of Lean's Type System

Wietze Koops and Niels Vooijs

December 7, 2021

Introduction

Focus on differences between Lean and Coq.

Three negative results, two positive results:

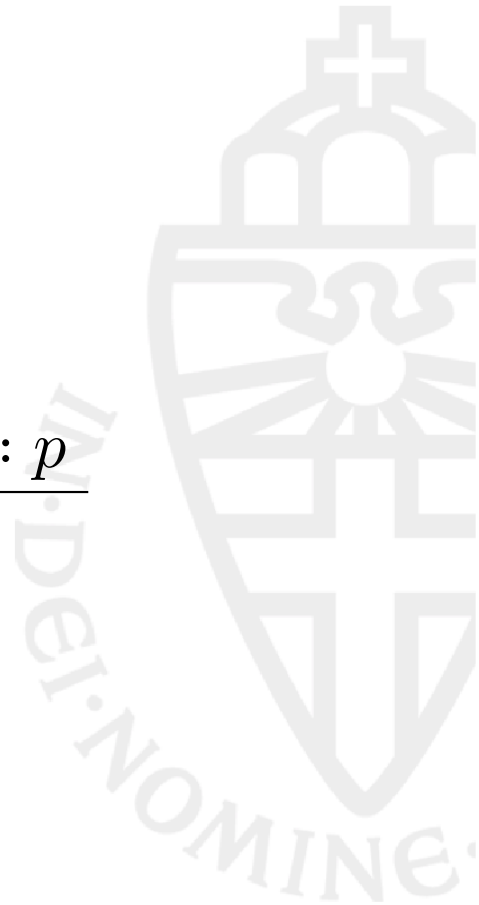
- Undecidability of definitional equality
- Algorithmic equality is not transitive
- Failure of subject reduction
- Unique typing
- Church-Rosser property

Recap

- Definitional equality $\Gamma \vdash e \equiv e'$
- Notable difference compared to Coq:
  proof irrelevance rule

$$\frac{\Gamma \vdash p : \mathbb{P} \qquad \Gamma \vdash h : p \qquad \Gamma \vdash h' : p}{\Gamma \vdash h \equiv h'}$$

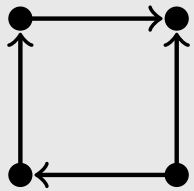
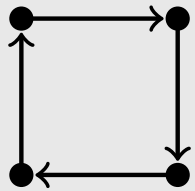
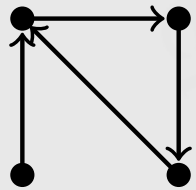
- Algorithmic equality $\Gamma \vdash e \Leftrightarrow e'$



Accessibility relation

$$\alpha : \mathsf{U}_\ell,\ R : \alpha \to \alpha \to \mathbb{P} \vdash \mathsf{acc}_R :=$$
$$\mu A : \alpha \to \mathbb{P}.(\mathsf{intro} : \forall x : \alpha.\ (\forall y : \alpha.\ R\ y\ x \to A\ y) \to A\ x)$$

Example

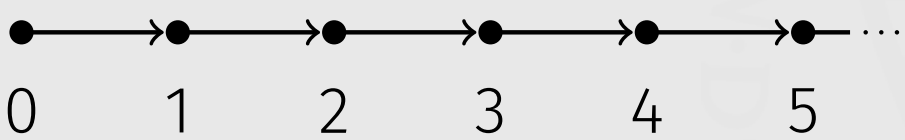




Accessibility relation

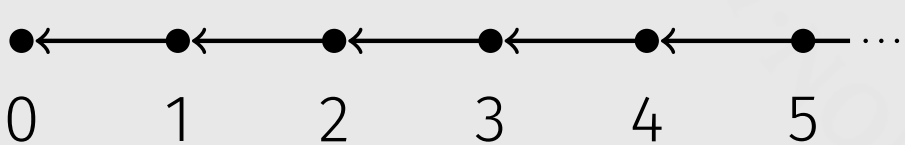
$$\alpha : \mathsf{U}_\ell,\ R : \alpha \to \alpha \to \mathbb{P} \vdash \mathsf{acc}_R :=$$
$$\mu A : \alpha \to \mathbb{P}.(\mathsf{intro} : \forall x : \alpha.\ (\forall y : \alpha.\ R\ y\ x \to A\ y) \to A\ x)$$

Example

$\forall n[\mathsf{acc}_<\ n]$



$\forall n[\neg\mathsf{acc}_>\ n]$



Note: taking the transitive closure does not matter for acc.

Undecidability of Definitional Equality

- Definitional equality is undecidable
- Family $P_e : \mathbb{N} \to \mathbf{2}$ s.t.
  - $P_e(n)$ decidable in e
  - $\exists n[P_e(n)]$ undecidable in e
- E.g. $P_e(n) :=$ "a TM $M_e(e)$ halts in n steps"
- $\exists n[P_e(n)]$: $M_e(e)$ halts $\implies$ Halting problem

Recursion on $\mathbb{N}$

- $n : \mathbb{N}$
- Decide $P\,n$ by if-then-else
- Multiple n by recursion

- $\text{acc}_>$ for $(\mathbb{N}, >)$
- Induction principle allows infinitely deep recursion

- Define $f : \forall n : \mathbb{N}[\text{acc}_>\ n \to \mathbf{1}]$ s.t.
- $f\,n\,(\text{intro}_{\text{acc}}\ n\ g) \equiv \text{if } P\,n \text{ then } () \text{ else } f\,(n+1)\,(\ldots)$
  - $(\ldots) = (g\,(n+1)\,p_n) : \text{acc}_>(n+1)$
  - $p_n : n < n+1$

Recursion on $\mathbb{N}$

- $n : \mathbb{N}$
- Decide $P\,n$ by if-then-else
- Multiple n by recursion

- $\text{acc}_>$ for $(\mathbb{N}, >)$
- Induction principle allows infinitely deep recursion

- Define $f : \forall n : \mathbb{N}[\text{acc}_>\ n \to \mathbf{1}]$ s.t.
- $f\,n\,(\text{intro}_{\text{acc}}\ n\ g) \equiv \text{if } P\,n \text{ then } () \text{ else } f\,(n+1)\,(\ldots)$
  - $(\ldots) = (g\,(n+1)\,p_n) : \text{acc}_>(n+1)$
  - $p_n : n < n+1$

Recursion on $\mathbb{N}$

- $n : \mathbb{N}$
- Decide $P\,n$ by if-then-else
- Multiple n by recursion

- $\text{acc}_>$ for $(\mathbb{N}, >)$
- Induction principle allows infinitely deep recursion

- Define $f : \forall n : \mathbb{N}[\text{acc}_> \; n \to \mathbf{1}]$ s.t.
- $f\,n\,(\text{intro}_{\text{acc}}\; n\; g) \equiv \text{if } P\,n \text{ then } () \text{ else } f\,(n+1)\,(\dots)$
  - $(\dots) = (g\,(n+1)\,p_n) : \text{acc}_>(n+1)$
  - $p_n : n < n+1$

Definition of f in Coq

```
Fixpoint f (n : nat) (H : acc nat gt n) : True :=
  match H with
  | acc_intro _ _ n g =>
    if P n then I
    else f (S n) (g (S n) (le_n (S n)))
  end.
```

Definition of f in Lean

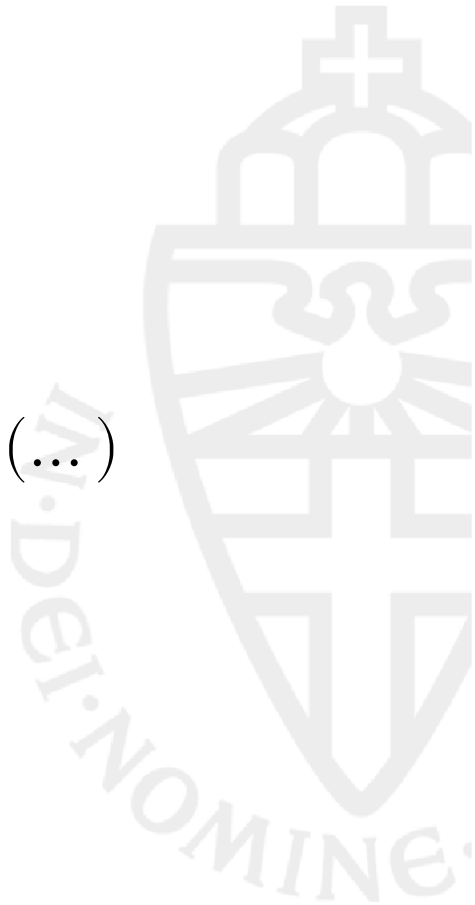
```
variable P : nat → bool
include P

def f : ∀ n : nat, acc gt n → unit :=
begin
  intros n H,
  induction H with n _ g,
  let p := show (n < n + 1), by constructor,
  exact (if P n then () else g (n + 1) p)
end
```

Applying f

- Assume $a : \mathrm{acc}_{>}\ n$
- We would like

$$f\ n\ a \equiv \text{if } P\,n \text{ then } () \text{ else } f\ (n+1)\ (\ldots)$$

- But that requires a of the form $\mathrm{intro}_{\mathrm{acc}}$ ...



Properties of acc

- $\mathsf{acc}_R\ x \to \forall y : \alpha[R\ y\ x \to \mathsf{acc}_R\ y]$
- The only way to prove $\mathsf{acc}_R\ x$ is to prove $\forall y : \alpha.R\ y\ x \to \mathsf{acc}_R\ y$
- Formally, prove by induction on the definition of acc_R
- Call the proof inv_x

Using Definitional Proof Irrelevance

- $\mathsf{inv}_x : \mathsf{acc}_R\ x \to \forall y : \alpha[R\ y\ x \to \mathsf{acc}_R\ y]$
- Assume $a : \mathsf{acc}_R\ x$
- Then $\mathsf{intro}_{\mathsf{acc}}\ x\ (\mathsf{inv}_x\ a) : \mathsf{acc}_R\ x$
- By propositional proof irrelevance

$$\frac{\Gamma \vdash \mathsf{acc}_R\ x : \mathbb{P} \qquad \Gamma \vdash a : \mathsf{acc}_R\ x \qquad \Gamma \vdash \mathsf{intro}_{\mathsf{acc}}\ x\ (\mathsf{inv}_x\ a) : \mathsf{acc}_R\ x}{\Gamma \vdash a \equiv \mathsf{intro}_{\mathsf{acc}}\ x\ (\mathsf{inv}_x\ a)}$$

Deciding $\exists n[P(n)]$

- Assume $a : \mathsf{acc}_{>}\ n$
- Work out $f\ n\ a$

$$\begin{aligned} f\ n\ a &\equiv f\ n\ (\mathsf{intro}_{\mathsf{acc}}\ n\ (\mathsf{inv}_n\ a)) \\ &\equiv \text{if } P\,n \text{ then } () \text{ else } f\ (n+1)\ (\ldots) \end{aligned}$$

  - Where $(\ldots) = (\mathsf{inv}_n\ a\ (n+1)\ p_n)$
  - $(\ldots) : \mathsf{acc}_{>}\ (n+1)$
- $a : \mathsf{acc}_{>} 0 \vdash f\ 0\ a \equiv ()$ iff $\exists n[P(n)]$

Non-transitivity of $\Leftrightarrow$

- $\equiv$ cannot be implemented
- introduce decidable $\Leftrightarrow$
- Not transitive
- Still

$$f\ n\ a \Leftrightarrow f\ n\ (\text{intro}_{\text{acc}}\ n\ (\text{inv}_n\ a))$$
$$\Leftrightarrow \text{if } P\,n \text{ then } () \text{ else } f\ (n+1)\ (\ldots)$$

but not

$$f\ n\ a \Leftrightarrow \text{if } P\,n \text{ then } () \text{ else } f\ (n+1)\ (\ldots)$$

Non-transitivity of $\Leftrightarrow$

- $\equiv$ cannot be implemented
- introduce decidable $\Leftrightarrow$
- Not transitive
- Still

$$f\ n\ a \Leftrightarrow f\ n\ (\text{intro}_{\text{acc}}\ n\ (\text{inv}_n\ a))$$
$$\Leftrightarrow \text{if } P\,n \text{ then } () \text{ else } f\ (n+1)\ (\dots)$$

but not

$$f\ n\ a \Leftrightarrow \text{if } P\,n \text{ then } () \text{ else } f\ (n+1)\ (\dots)$$

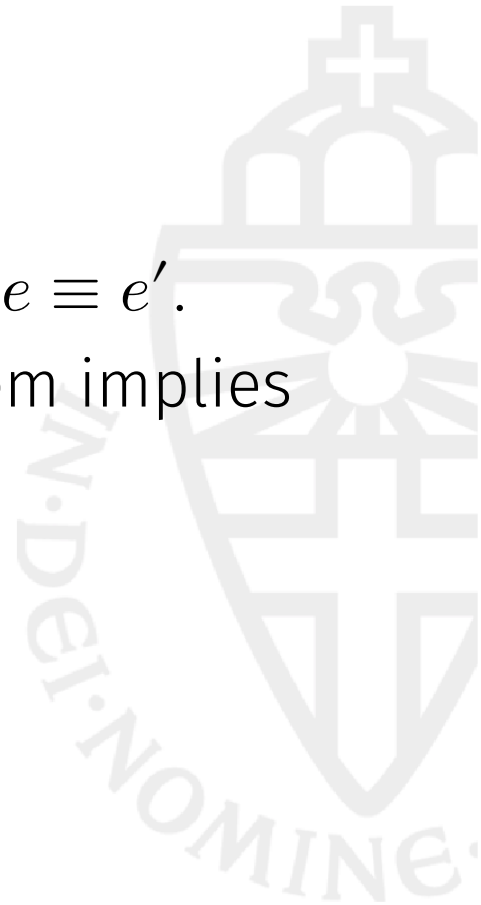
Failure of Subject Reduction

- Due to the non-transitivity
- Assume $\Gamma \vdash \alpha \Leftrightarrow \beta, \beta \Leftrightarrow \gamma$ and $\Gamma \nvdash \alpha \Leftrightarrow \gamma$
- Assume $\Gamma \vdash e : \gamma$
- $\Gamma \vdash_{\mathrm{L}} \mathrm{id}_{\beta}\ e : \alpha$ by checking $\alpha \Leftrightarrow \beta$ and $\beta \Leftrightarrow \gamma$
- $\Gamma \nvdash_{\mathrm{L}} e : \alpha$ since $\alpha \not\Leftrightarrow \gamma$

- Also shows how to circumvent failing type judgements due to use of $\Leftrightarrow$ instead of $\equiv$

Algorithmic and definitional equality

- If $\Gamma \vdash_{\mathrm{L}} e : \alpha$, $e' : \alpha$ and $\Gamma \vdash_{\mathrm{L}} e \Leftrightarrow e'$, then $\Gamma \vdash e \equiv e'$.
- Showing consistency of the ideal typing system implies consistency of Lean



Unique typing

Theorem (Unique typing)

If $\Gamma \vdash e : \alpha$ *and* $\Gamma \vdash e : \beta$*, then* $\Gamma \vdash \alpha \equiv \beta$

- Does not hold in Coq because of universe cumulativity: a universe U_i can be replaced by a universe U_j where $i \leq j$.

Unique typing

Theorem (Unique typing)

If $\Gamma \vdash e : \alpha$ *and* $\Gamma \vdash e : \beta$*, then* $\Gamma \vdash \alpha \equiv \beta$

Proof idea 1: Make relations $\vdash_n$ that grow to $\vdash$ as $n \to \infty$:

- $\Gamma \vdash_0 \alpha \equiv \beta$ iff $\alpha = \beta$
- Assume $\Gamma \vdash_m \alpha \equiv \beta$ is defined for $m \leq n$. Then $\Gamma \vdash_n e : \alpha$ iff there is a proof of $\Gamma \vdash e : \alpha$ in which all appeals to the conversion rule have the form $\Gamma \vdash_m \alpha \equiv \beta$ for $m \leq n$
- $\Gamma \vdash_{n+1} \alpha \equiv \beta$ iff there is a proof of $\Gamma \vdash \alpha \equiv \beta$ using only $\Gamma \vdash_n e : \alpha$ type judgements

Unique typing

Definition (Definitional inversion)

We say that $\vdash_n$ has definitional inversion if the following hold:

- If $\Gamma \vdash_n \mathsf{U}_\ell \equiv \mathsf{U}_{\ell'}$, then $\ell \equiv \ell'$.
- If $\Gamma \vdash_n \forall x : \alpha.\beta \equiv \forall x : \alpha'.\beta'$, then $\Gamma \vdash_n \alpha \equiv \alpha'$ and $\Gamma, x : \alpha \vdash_n \beta \equiv \beta'$
- $\Gamma \vdash_n \mathsf{U}_\ell \not\equiv \forall x : \alpha.\beta$

Proof idea 2: show that $\vdash_n$ has definitional inversion and that definitional inversion implies unique typing.

Church-Rosser property

- Church-Rosser property in Coq: If $\Gamma \vdash e : \alpha$ and $e \rightsquigarrow^* e_1$ and $e \rightsquigarrow^* e_2$, then there exist e_1' and e_2' such that $e_1' = e_2'$ and $e_1 \rightsquigarrow^* e_1'$ and $e_2 \rightsquigarrow^* e_2'$.
- Does not hold in Lean: then normal forms would be unique
- Need proof equivalence relation $\equiv_p$ describing in which ways normal forms fail to be unique
- $\equiv_p$ is a 'small' relation

Church-Rosser property

Theorem (Church-Rosser)

If $\Gamma \vdash e : \alpha$ *and* $e \rightsquigarrow_\kappa^* e_1$ *and* $e \rightsquigarrow_\kappa^* e_2$*, then there exist* e_1' *and* e_2' *such that* $\Gamma \vdash e_1' \equiv_p e_2'$ *and* $e_1 \rightsquigarrow_\kappa^* e_1'$ *and* $e_2 \rightsquigarrow_\kappa^* e_2'$*.*

- To complete the proof, we also need to strengthen $\rightsquigarrow$ to $\rightsquigarrow_\kappa$

Church-Rosser property

Theorem (Church-Rosser)

If $\Gamma \vdash e : \alpha$ *and* $e \twoheadrightarrow_\kappa^* e_1$ *and* $e \twoheadrightarrow_\kappa^* e_2$*, then there exist* e_1' *and* e_2' *such that* $\Gamma \vdash e_1' \equiv_p e_2'$ *and* $e_1 \twoheadrightarrow_\kappa^* e_1'$ *and* $e_2 \twoheadrightarrow_\kappa^* e_2'$*.*

Proof ideas:

- Similar to proof for β-reduction
- Parallel reduction relation $\gg_\kappa$
- Strong Diamond Property: * is replaced by relation $\ggg_\kappa$

Finishing the proof

- Define that $\Gamma \vdash e_1 \equiv_\kappa e_2$ iff $\Gamma \vdash e_1, e_2 : \alpha$ for some α, and there exist e_1', e_2' such that $\Gamma \vdash e_1 \rightsquigarrow_\kappa^* e_1'$ and $\Gamma \vdash e_2 \rightsquigarrow_\kappa^* e_2'$ and $\Gamma \vdash e_1 \equiv_p e_2$
- Church-Rosser implies transitivity of $\equiv_\kappa$
- Then $\equiv_\kappa$ and $\equiv$ are the same
- This 'splits up' definitional equivalence in $\equiv_p$ and $\rightsquigarrow_\kappa^*$ which allows proving definitional inversion of $\vdash_n$.

Conclusion

- Four desirable properties of an equality relation

relation	decidability	transitivity	β-reduction	proof irrelevance
$\equiv$	✗	✓	✓	✓
$\Leftrightarrow$	✓	✗	✓	✓
$\equiv_p$	?	✓	✗	✓
Coq	✓	✓	✓	✗