A Novice-Friendly Induction Tactic for Lean (Short Talk)

Jannis Limperg

Vrije Universiteit Amsterdam

CPP 2021

Simplification of Induction Hypotheses

[Conor McBride, Elimination with a Motive, TYPES 2000]

Simplification of Induction Hypotheses

```
∀ S′ s′,
  (while (λ _, true) S′, s′) =
  (while (λ _, true) S, t) →
  false
```

[Conor McBride, Elimination with a Motive, TYPES 2000]

Simplification of Induction Hypotheses

```
∀ S′ s′,
  (while (λ _, true) S′, s′) =
  (while (λ _, true) S, t) →
  false
```

```
(while (λ _, true) ?S′, ?s′) =
(while (λ _, true) S,  t) →
false
```

[Conor McBride, Elimination with a Motive, TYPES 2000]

Simplification of Induction Hypotheses

```
∀ S′ s′,
  (while (λ _, true) S′, s′) =
  (while (λ _, true) S, t) →
  false
```

```
(while (λ _, true) ?S′, ?s′) =
(while (λ _, true) S,  t) →
false
```

```
(while (λ _, true) S, t) =
(while (λ _, true) S, t) →
false
```

[Conor McBride, Elimination with a Motive, TYPES 2000]

Simplification of Induction Hypotheses

```
∀ S′ s′,
  (while (λ _, true) S′, s′) =
  (while (λ _, true) S, t) →
  false
```

```
(while (λ _, true) ?S′, ?s′) =
(while (λ _, true) S,  t) →
false
```

```
(while (λ _, true) S, t) =
(while (λ _, true) S, t) →
false
```

```
false
```

[Conor McBride, Elimination with a Motive, TYPES 2000]

Automatic Generalisation of Induction Hypotheses

```
∀ n m, n + n = m + m → n = m
```

[Benjamin C. Pierce et al., Software Foundations]

Automatic Generalisation of Induction Hypotheses

```
∀ n m, n + n = m + m → n = m
```

```
…
ih : n + n = m + m → n = m
h : n + n + 2 = m + m
⊢ n + 1 = m
```

[Benjamin C. Pierce et al., Software Foundations]

Automatic Generalisation of Induction Hypotheses

```
∀ n m, n + n = m + m → n = m
```

```
…
ih : n + n = m + m → n = m
h : n + n + 2 = m + m
⊢ n + 1 = m
```

```
…
ih : ∀ m, n + n = m + m → n = m
h : n + n + 2 = m + m
⊢ n + 1 = m
```

[Benjamin C. Pierce et al., Software Foundations]

Naming

Naming

```
∀ α (r : α → α → Type) (a b c : α)
  (h₁ : tc r a b) (h₂ : tc r b c), tc r a c
```

Naming

```
∀ α (r : α → α → Type) (a b c : α)
  (h₁ : tc r a b) (h₂ : tc r b c), tc r a c
```

```
α : Type
r : α → α → Type
c a b h₁_x h₁_y h₁_z : α
h₁_hr : r h₁_x h₁_y
h₁_ht : tc r h₁_y h₁_z
h₁_ih : tc r h₁_z c → tc r h₁_y c
h₂ : tc r h₁_z c
⊢ tc r h₁_x c
```

Naming

```
∀ α (r : α → α → Type) (a b c : α)
  (h₁ : tc r a b) (h₂ : tc r b c), tc r a c
```

```
α : Type
r : α → α → Type
a y b c : α
hr : r a y
h₁ : tc r y b
ih : ∀ c, tc r b c → tc r y c
h₂ : tc r b c
⊢ tc r a c
```

Summary

- Simplification of induction hypotheses
- Automatic generalisation of induction hypotheses
- Naming
- (Not in this talk:) Evaluation of Lean 3's metaprogramming framework

Jannis Limperg
j.b.limperg@vu.nl
`https://limperg.de/contact`