Uniform Inductive Reasoning in Transitive Closure Logic via Infinite Descent

Liron Cohen [1] Reuben N. S. Rowe [2]

SREPLS 10, Birkbeck, University of London

Tuesday 18th September 2018

[1]Dept of Computer Science, Cornell University, Ithaca, NY, USA

[2]School of Computing, University of Kent, Canterbury, UK

Motivation

- Carry out formal inductive reasoning
- Do so automatically (as much as possible)
- Study/compare different ‘styles’ of inductive reasoning

Formalising Inductive Reasoning

Explicit Inductive Definitions

- Use clauses to inductively define predicates:

$$\phi_1 \wedge \ldots \wedge \phi_n \Rightarrow P(\vec{t})$$
$$\vdots$$
$$\psi_1 \wedge \ldots \wedge \psi_m \Rightarrow P(\vec{t})$$

- We take the smallest interpretation closed under the rules

$$\frac{}{\mathsf{N}\,0} \qquad \frac{\mathsf{N}\,x}{\mathsf{N}\,\mathsf{s}x} \qquad \frac{}{\mathsf{E}\,0} \qquad \frac{\mathsf{O}\,x}{\mathsf{E}\,\mathsf{s}x} \qquad \frac{\mathsf{E}\,x}{\mathsf{O}\,\mathsf{s}x}$$

$$[\![\mathsf{N}]\!] = \{0, \mathsf{s}0, \mathsf{ss}0, \ldots, \mathsf{s}^n\,0, \ldots\}$$
$$[\![\mathsf{E}]\!] = \{0, \mathsf{ss}0, \ldots, \mathsf{s}^{2n}\,0, \ldots\}$$
$$[\![\mathsf{O}]\!] = \{\mathsf{s}0, \ldots, \mathsf{s}^{2n+1}\,0, \ldots\}$$

Reasoning Using Explicit Induction Principles

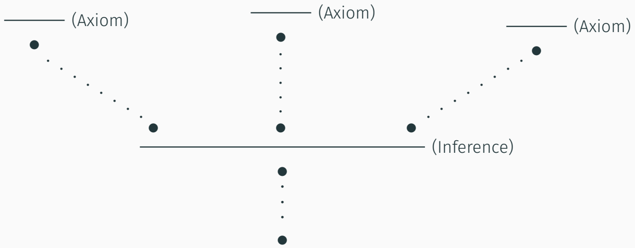
- We reason using the corresponding induction principles

$$\frac{\Gamma \vdash \mathrm{IND}_Q(F)\ (\forall Q \text{ mutually recursive with } P) \qquad \Gamma, F(\vec{t}) \vdash \Delta}{\Gamma, \mathrm{P}\,\vec{t} \vdash \Delta}$$

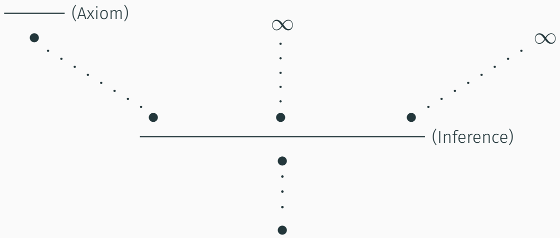
- E.g. the productions for **N** give

$$\frac{\Gamma \vdash F(\mathbf{0}) \quad \Gamma, F(x) \vdash F(\mathbf{s}x) \quad \Gamma, F(t) \vdash \Delta}{\Gamma, \mathbf{N}t \vdash \Delta}$$

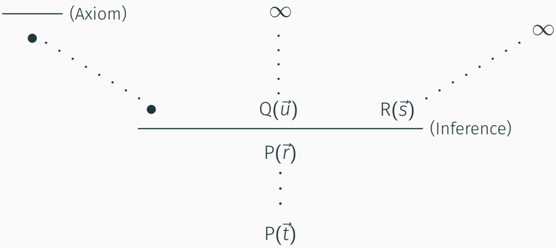
Non-well-founded Proofs: Reasoning by Infinite Descent



Non-well-founded Proofs: Reasoning by Infinite Descent

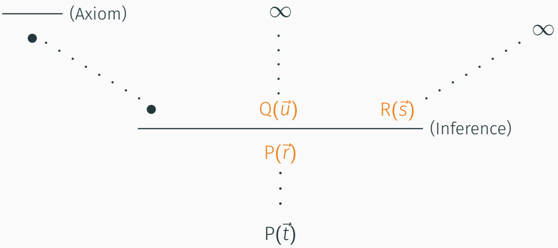


Non-well-founded Proofs: Reasoning by Infinite Descent



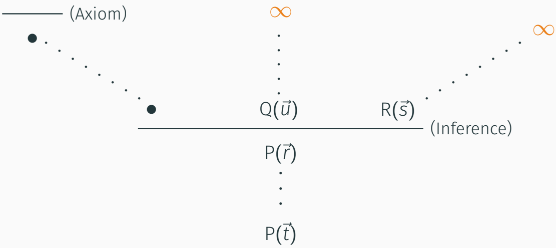
- We trace predicate instances through the proof

Non-well-founded Proofs: Reasoning by Infinite Descent



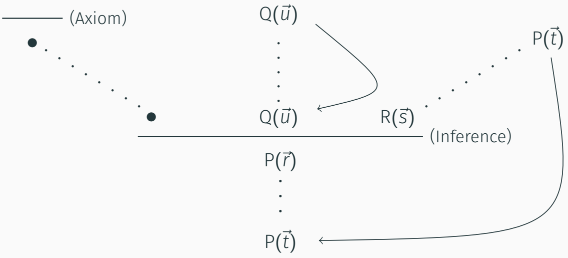
- We trace predicate instances through the proof
    - At certain points, these progress (i.e. get 'smaller')

Non-well-founded Proofs: Reasoning by Infinite Descent



- We trace predicate instances through the proof
  - At certain points, these progress (i.e. get ‘smaller’)
- Each infinite path must admit some infinite descent

Non-well-founded Proofs: Reasoning by Infinite Descent



- We trace predicate instances through the proof
    - At certain points, these progress (i.e. get 'smaller')
- Each infinite path must admit some infinite descent
- This global trace condition is an ω-regular property
    - i.e. decidable using Büchi automata

An Example Cyclic Proof

$$\Rightarrow \; \mathrm{N}\,\mathbf{0}$$

$$\mathrm{N}\,x \Rightarrow \; \mathrm{N}\,\mathbf{s}x$$

$$\Rightarrow \; \mathrm{E}\,\mathbf{0}$$

$$\mathrm{O}\,x \Rightarrow \; \mathrm{E}\,\mathbf{s}x$$

$$\mathrm{E}\,x \Rightarrow \; \mathrm{O}\,\mathbf{s}x$$

$$\dfrac{\dfrac{\dfrac{}{\vdash \mathrm{N}\,\mathbf{0}}\,(\mathrm{N}\,\mathrm{R}_1)}{x = \mathbf{0} \vdash \mathrm{N}\,x}\,(=\mathrm{L}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathrm{E}\,x \vdash \mathrm{N}\,x}{\mathrm{E}\,z \vdash \mathrm{N}\,z}\,(\mathrm{Subst})}{\mathrm{E}\,z \vdash \mathrm{N}\,\mathbf{s}z}\,(\mathrm{N}\,\mathrm{R}_2)}{y = \mathbf{s}z,\ \mathrm{E}\,z \vdash \mathrm{N}\,y}\,(=\mathrm{L})}{\mathrm{O}\,y \vdash \mathrm{N}\,y}\,(\mathrm{Case}\ \mathrm{O})}{\mathrm{O}\,y \vdash \mathrm{N}\,\mathbf{s}y}\,(\mathrm{N}\,\mathrm{R}_2)}{x = \mathbf{s}y,\ \mathrm{O}\,y \vdash \mathrm{N}\,x}\,(=\mathrm{L})}{\mathrm{E}\,x \vdash \mathrm{N}\,x}\,(\mathrm{Case}\ \mathrm{E})$$

An Example Cyclic Proof

Left unfolding rule

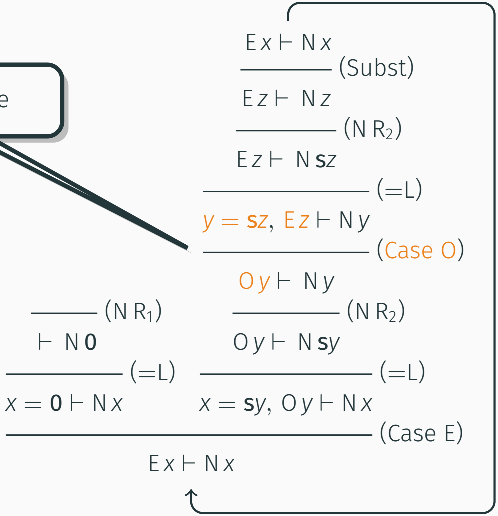
$$\Rightarrow \mathsf{N}\,\mathbf{0}$$
$$\mathsf{N}\,x \Rightarrow \mathsf{N}\,\mathsf{s}x$$
$$\Rightarrow \mathsf{E}\,\mathbf{0}$$
$$\mathsf{O}\,x \Rightarrow \mathsf{E}\,\mathsf{s}x$$
$$\mathsf{E}\,x \Rightarrow \mathsf{O}\,\mathsf{s}x$$

$$\dfrac{\dfrac{\dfrac{}{\vdash \mathsf{N}\,\mathbf{0}}\;(\mathsf{N}\,\mathsf{R}_1)}{x = \mathbf{0} \vdash \mathsf{N}\,x}\;(=\mathsf{L}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{E}\,x \vdash \mathsf{N}\,x}{\mathsf{E}\,z \vdash \mathsf{N}\,z}\;(\mathsf{Subst})}{\mathsf{E}\,z \vdash \mathsf{N}\,\mathsf{s}z}\;(\mathsf{N}\,\mathsf{R}_2)}{y = \mathsf{s}z,\ \mathsf{E}\,z \vdash \mathsf{N}\,y}\;(=\mathsf{L})}{\mathsf{O}\,y \vdash \mathsf{N}\,y}\;(\mathsf{Case}\ \mathsf{O})}{\mathsf{O}\,y \vdash \mathsf{N}\,\mathsf{s}y}\;(\mathsf{N}\,\mathsf{R}_2)}{x = \mathsf{s}y,\ \mathsf{O}\,y \vdash \mathsf{N}\,x}\;(=\mathsf{L})}{\mathsf{E}\,x \vdash \mathsf{N}\,x}\;(\mathsf{Case}\ \mathsf{E})$$

An Example Cyclic Proof

Left unfolding rule

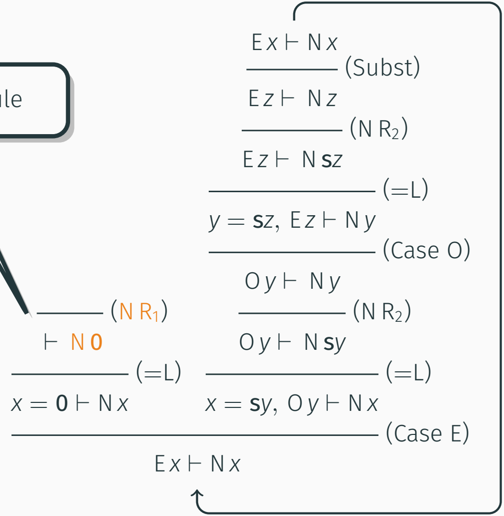
$$\Rightarrow \mathrm{N}\,\mathbf{0}$$
$$\mathrm{N}\,x \Rightarrow \mathrm{N}\,\mathsf{s}x$$
$$\Rightarrow \mathrm{E}\,\mathbf{0}$$
$$\mathrm{O}\,x \Rightarrow \mathrm{E}\,\mathsf{s}x$$
$$\mathrm{E}\,x \Rightarrow \mathrm{O}\,\mathsf{s}x$$



An Example Cyclic Proof

Right unfolding rule

$$\Rightarrow \mathrm{N}\,\mathbf{0}$$
$$\mathrm{N}\,x \Rightarrow \mathrm{N}\,\mathbf{s}x$$
$$\Rightarrow \mathrm{E}\,\mathbf{0}$$
$$\mathrm{O}\,x \Rightarrow \mathrm{E}\,\mathbf{s}x$$
$$\mathrm{E}\,x \Rightarrow \mathrm{O}\,\mathbf{s}x$$



An Example Cyclic Proof

Right unfolding rule

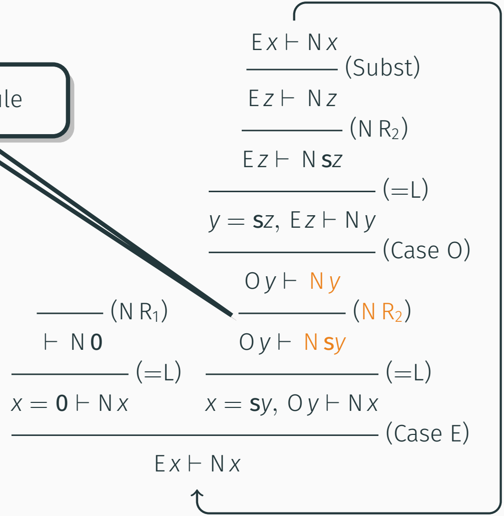
$$\Rightarrow \mathsf{N}\,\mathbf{0}$$

$$\mathsf{N}\,x \Rightarrow \mathsf{N}\,\mathsf{s}x$$

$$\Rightarrow \mathsf{E}\,\mathbf{0}$$

$$\mathsf{O}\,x \Rightarrow \mathsf{E}\,\mathsf{s}x$$

$$\mathsf{E}\,x \Rightarrow \mathsf{O}\,\mathsf{s}x$$



An Example Cyclic Proof

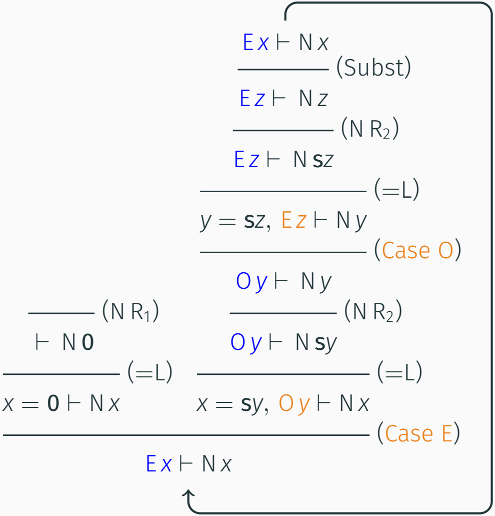
Right unfolding rule

$$\begin{aligned} &\Rightarrow \mathsf{N}\,\mathbf{0} \\ \mathsf{N}\,x &\Rightarrow \mathsf{N}\,\mathbf{s}x \\ &\Rightarrow \mathsf{E}\,\mathbf{0} \\ \mathsf{O}\,x &\Rightarrow \mathsf{E}\,\mathbf{s}x \\ \mathsf{E}\,x &\Rightarrow \mathsf{O}\,\mathbf{s}x \end{aligned}$$

$$\dfrac{\dfrac{\dfrac{}{\vdash \mathsf{N}\,\mathbf{0}}\,(\mathsf{N}\,\mathrm{R}_1)}{x = \mathbf{0} \vdash \mathsf{N}\,x}\,(=\mathrm{L}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{E}\,x \vdash \mathsf{N}\,x}{\mathsf{E}\,z \vdash \mathsf{N}\,z}\,(\mathrm{Subst})}{\mathsf{E}\,z \vdash \mathsf{N}\,\mathbf{s}z}\,(\mathsf{N}\,\mathrm{R}_2)}{y = \mathbf{s}z,\ \mathsf{E}\,z \vdash \mathsf{N}\,y}\,(=\mathrm{L})}{\mathsf{O}\,y \vdash \mathsf{N}\,y}\,(\mathrm{Case}\ \mathsf{O})}{\mathsf{O}\,y \vdash \mathsf{N}\,\mathbf{s}y}\,(\mathsf{N}\,\mathrm{R}_2)}{x = \mathbf{s}y,\ \mathsf{O}\,y \vdash \mathsf{N}\,x}\,(=\mathrm{L})}{\mathsf{E}\,x \vdash \mathsf{N}\,x}\,(\mathrm{Case}\ \mathsf{E})$$

An Example Cyclic Proof

$$
\begin{aligned}
&\Rightarrow \mathsf{N}\,\mathbf{0} \\
\mathsf{N}\,x &\Rightarrow \mathsf{N}\,\mathsf{s}x \\
&\Rightarrow \mathsf{E}\,\mathbf{0} \\
\mathsf{O}\,x &\Rightarrow \mathsf{E}\,\mathsf{s}x \\
\mathsf{E}\,x &\Rightarrow \mathsf{O}\,\mathsf{s}x
\end{aligned}
$$



Comparing the Two Approaches

For FOL with Martin-Löf style inductive definitions:

[Brotherston & Simpson, 2007]

- Infinitary system sound/complete for standard semantics

- Explicit induction sound/complete for Henkin semantics

Comparing the Two Approaches

For FOL with Martin-Löf style inductive definitions:

[Brotherston & Simpson, 2007]

- Infinitary system sound/complete for standard semantics
- Cyclic system subsumes explicit induction
- Explicit induction sound/complete for Henkin semantics

Comparing the Two Approaches

For FOL with Martin-Löf style inductive definitions:

[Brotherston & Simpson, 2007]

- Infinitary system sound/complete for standard semantics
- Cyclic system subsumes explicit induction
  - Equivalent under arithmetic
  - Not equivalent in general (2-Hydra counterexample)

[Berardi & Tatsuta, 2017]

- Explicit induction sound/complete for Henkin semantics

Transitive Closure Logic

Transitive Closure Logic

Transitive Closure (**TC**) Logic extends FOL with formulas:

- $(RTC_{x,y}\,\varphi)(s,t)$
  - φ is a formula
  - x and y are distinct variables (which become bound in φ)
  - s and t are terms

whose intended meaning is an infinite disjunction

$$
\begin{aligned}
s = t &\vee \varphi[s/x, t/y] \\
&\vee (\exists w_1 \,.\, \varphi[s/x, w_1/y] \wedge \varphi[w_1/x, t/y]) \\
&\vee (\exists w_1, w_2 \,.\, \varphi[s/x, w_1/y] \wedge \varphi[w_1/x, w_2/y] \wedge \varphi[w_2/x, t/y]) \\
&\vee \ \ldots
\end{aligned}
$$

Transitive Closure Logic: Standard Semantics

The formal semantics:

- M is a (standard) first-order model with domain D
- v is a valuation of terms in M:

$$M, v \models (RTC_{x,y}\,\varphi)(s, t)$$

Transitive Closure Logic: Standard Semantics

The formal semantics:

- M is a (standard) first-order model with domain D
- v is a valuation of terms in M:

$$M, v \models (RTC_{x,y}\,\varphi)(s,t) \Leftrightarrow$$
$$\exists a_0, \ldots, a_n \in D$$

Transitive Closure Logic: Standard Semantics

The formal semantics:

- M is a (standard) first-order model with domain D
- v is a valuation of terms in M:

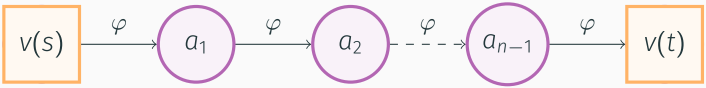
$$M, v \models (RTC_{x,y}\, \varphi)(s, t) \Leftrightarrow$$
$$\exists a_0, \ldots, a_n \in D \,.\, v(s) = a_0 \wedge v(t) = a_n$$

Transitive Closure Logic: Standard Semantics

The formal semantics:

- M is a (standard) first-order model with domain D
- v is a valuation of terms in M:

$$
\begin{aligned}
M, v \models (RTC_{x,y}\,\varphi)(s,t) \Leftrightarrow \\
&\exists a_0, \ldots, a_n \in D \,.\, v(s) = a_0 \wedge v(t) = a_n \\
&\quad \wedge M, v[x := a_i, y := a_{i+1}] \models \varphi \quad \text{for all } i < n
\end{aligned}
$$



Example: Arithmetic in TC

Take a signature $\Sigma = \{\mathbf{0}, \mathbf{s}\}$ + equality

$$\mathsf{Nat}(x) \equiv (RTC_{v,w}\, \mathbf{s}v = w)(\mathbf{0}, x)$$

Example: Arithmetic in TC

Take a signature $\Sigma = \{\mathbf{0}, \mathsf{s}\}$ + equality

$$\mathsf{Nat}(x) \equiv (RTC_{v,w}\, \mathsf{s}v = w)(\mathbf{0}, x)$$

$$x \leq y \equiv (RTC_{v,w}\, \mathsf{s}v = w)(x, y)$$

Example: Arithmetic in TC

Take a signature $\Sigma = \{\mathsf{0}, \mathsf{s}\}$ + equality and pairing

$$\mathsf{Nat}(x) \equiv (RTC_{v,w}\, \mathsf{s}v = w)(\mathsf{0}, x)$$

$$x \leq y \equiv (RTC_{v,w}\, \mathsf{s}v = w)(x, y)$$

"$x = y + z$" $\equiv$

$$(RTC_{v,w}\, \exists n_1, n_2\, .\, v = \langle n_1, n_2 \rangle \wedge w = \langle \mathsf{s}n_1, \mathsf{s}n_2 \rangle)(\langle \mathsf{0}, y \rangle, \langle z, x \rangle)$$

Example: Arithmetic in TC

Take a signature $\Sigma = \{\mathbf{0}, \mathbf{s}\}$ + equality and pairing

$$\mathsf{Nat}(x) \equiv (RTC_{v,w}\, \mathbf{s}v = w)(\mathbf{0}, x)$$

$$x \leq y \equiv (RTC_{v,w}\, \mathbf{s}v = w)(x, y)$$

"$x = y + z$" $\equiv$

$$(RTC_{v,w}\, \exists n_1, n_2\, .\, v = \langle n_1, n_2 \rangle \wedge w = \langle \mathbf{s}n_1, \mathbf{s}n_2 \rangle)(\langle \mathbf{0}, y \rangle, \langle z, x \rangle)$$

$\langle \mathbf{0}, y \rangle$

Example: Arithmetic in TC

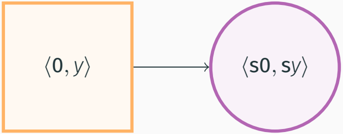
Take a signature $\Sigma = \{\mathbf{0}, \mathsf{s}\}$ + equality and pairing

$$\mathsf{Nat}(x) \equiv (RTC_{v,w}\, \mathsf{s}v = w)(\mathbf{0}, x)$$

$$x \leq y \equiv (RTC_{v,w}\, \mathsf{s}v = w)(x, y)$$

"$x = y + z$" $\equiv$

$$(RTC_{v,w}\, \exists n_1, n_2 \,.\, v = \langle n_1, n_2 \rangle \wedge w = \langle \mathsf{s}n_1, \mathsf{s}n_2 \rangle)(\langle \mathbf{0}, y \rangle, \langle z, x \rangle)$$



Example: Arithmetic in TC

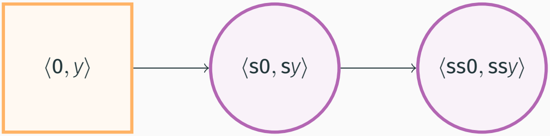
Take a signature $\Sigma = \{\mathsf{0}, \mathsf{s}\}$ + equality and pairing

$$\mathsf{Nat}(x) \equiv (RTC_{v,w}\, \mathsf{s}v = w)(\mathsf{0}, x)$$

$$x \leq y \equiv (RTC_{v,w}\, \mathsf{s}v = w)(x, y)$$

"$x = y + z$" $\equiv$

$$(RTC_{v,w}\, \exists n_1, n_2 \,.\, v = \langle n_1, n_2 \rangle \wedge w = \langle \mathsf{s}n_1, \mathsf{s}n_2 \rangle)(\langle \mathsf{0}, y \rangle, \langle z, x \rangle)$$



Example: Arithmetic in TC

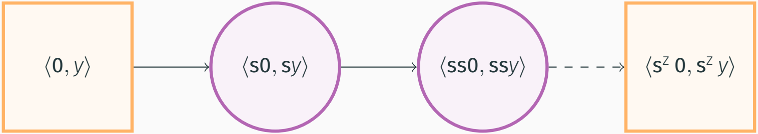
Take a signature $\Sigma = \{\mathbf{0}, \mathsf{s}\}$ + equality and pairing

$$\mathsf{Nat}(x) \equiv (RTC_{v,w}\,\mathsf{s}v = w)(\mathbf{0}, x)$$

$$x \leq y \equiv (RTC_{v,w}\,\mathsf{s}v = w)(x, y)$$

"$x = y + z$" $\equiv$

$$(RTC_{v,w}\,\exists n_1, n_2 \,.\, v = \langle n_1, n_2\rangle \wedge w = \langle \mathsf{s}n_1, \mathsf{s}n_2\rangle)(\langle \mathbf{0}, y\rangle, \langle z, x\rangle)$$



Proof Rules for Reasoning in TC

reflexivity
$$\frac{}{\vdash (RTC_{x,y}\,\varphi)(t,t)}$$

step
$$\frac{\Gamma \vdash \Delta, (RTC_{x,y}\,\varphi)(s,r) \quad \Gamma \vdash \Delta, \varphi[r/x, t/y]}{\Gamma \vdash \Delta, (RTC_{x,y}\,\varphi)(s,t)}$$

induction
$$\frac{\Gamma \vdash \Delta, \psi[s/x] \quad \Gamma, \psi(x), \varphi(x,y) \vdash \Delta, \psi[y/x] \quad \Gamma, \psi[t/x] \vdash \Delta}{\Gamma, (RTC_{x,y}\,\varphi)(s,t) \vdash \Delta}$$

$x \notin \mathbf{fv}(\Gamma, \Delta)$ and $y \notin \mathbf{fv}(\Gamma, \Delta, \psi)$

Proof Rules for Reasoning in TC

reflexivity
$$\frac{}{\vdash (RTC_{x,y}\,\varphi)(t,t)}$$

step
$$\frac{\Gamma \vdash \Delta, (RTC_{x,y}\,\varphi)(s,r) \quad \Gamma \vdash \Delta, \varphi[r/x, t/y]}{\Gamma \vdash \Delta, (RTC_{x,y}\,\varphi)(s,t)}$$

induction
$$\frac{\Gamma \vdash \Delta, \psi[s/x] \quad \Gamma, \psi(x), \varphi(x,y) \vdash \Delta, \psi[y/x] \quad \Gamma, \psi[t/x] \vdash \Delta}{\Gamma, (RTC_{x,y}\,\varphi)(s,t) \vdash \Delta}$$

$x \notin \mathbf{fv}(\Gamma, \Delta)$ and $y \notin \mathbf{fv}(\Gamma, \Delta, \psi)$

case-split
$$\frac{\Gamma, s = t \vdash \Delta \quad \Gamma, (RTC_{x,y}\,\varphi)(s,z), \varphi[z/x, t/y] \vdash \Delta}{\Gamma, (RTC_{x,y}\,\varphi)(s,t) \vdash \Delta}\ (z \text{ fresh})$$

Proof Rules for Reasoning in TC

reflexivity
$$\frac{}{\vdash (RTC_{x,y}\,\varphi)(t,t)}$$

step
$$\frac{\Gamma \vdash \Delta, (RTC_{x,y}\,\varphi)(s,r) \quad \Gamma \vdash \Delta, \varphi[r/x, t/y]}{\Gamma \vdash \Delta, (RTC_{x,y}\,\varphi)(s,t)}$$

induction
$$\frac{\Gamma \vdash \Delta, \psi[s/x] \quad \Gamma, \psi(x), \varphi(x,y) \vdash \Delta, \psi[y/x] \quad \Gamma, \psi[t/x] \vdash \Delta}{\Gamma, (RTC_{x,y}\,\varphi)(s,t) \vdash \Delta}$$

$x \notin \mathbf{fv}(\Gamma, \Delta)$ and $y \notin \mathbf{fv}(\Gamma, \Delta, \psi)$

case-split
$$\frac{\Gamma, s = t \vdash \Delta \quad \Gamma, (RTC_{x,y}\,\varphi)(s,z), \varphi[z/x, t/y] \vdash \Delta}{\Gamma, (RTC_{x,y}\,\varphi)(s,t) \vdash \Delta}\ (z \text{ fresh})$$

Advantages of TC as a Formal Framework

- It is only a minimal extension of FOL
- It only requires a single, uniform induction principle
- No need to ‘choose’ particular inductive definitions
- It is a sufficiently expressive logic

Theorem (Avron ’03)

All finitely inductively definable relations[†] *are definable in* TC.

A. Avron, *Transitive Closure and the Mechanization of Mathematics.*

[†] as formalised in: S. Feferman, *Finitary Inductively Presented Logics*, 1989

Comparing Styles of Induction for TC

- Infinitary system sound/complete for standard semantics

- Explicit induction sound/complete for Henkin semantics

Comparing Styles of Induction for TC

- Infinitary system sound/complete for standard semantics
- Cyclic system subsumes explicit induction

- Explicit induction sound/complete for Henkin semantics

Comparing Styles of Induction for TC

- Infinitary system sound/complete for standard semantics
- Cyclic system subsumes explicit induction
    - Equivalent under arithmetic
- Explicit induction sound/complete for Henkin semantics

Comparing Styles of Induction for TC

- Infinitary system sound/complete for standard semantics
- Cyclic system subsumes explicit induction
  - Equivalent under arithmetic
  - **Don't know if they are inequivalent in general!**

    2-Hydra does not work since all inductive definitions available via *RTC*
- Explicit induction sound/complete for Henkin semantics

Future Work

- open question of equivalence for TC proof systems
- Implementation to support automated reasoning.
- Use TC to better study implicit vs explicit induction.
- Adapt TC for coinductive reasoning?

Thank you!