Induction, Transitive Closure and Cycles

Liron Cohen, Cornell University,
Reuben Rowe, University of Kent

ASL North American Annual Meeting, 2018
Applications of Logic in CS

Verification

Database

Complexity

Type Theory

Model Checking

Knowledge Reasoning

MKM

Inductive arguments on programs

Expressive Query languages (WITH RECURSIVE) SQL3, IBM DB2, Datalog

Characterization of complexity classes

Inductive definition of type judgments

Reachability properties

Common knowledge, defined inductively

Natural numbers

What Logic?

Halpern, Harper, Immerman, Kolaitis, Vardi, Vianu. On the unusual effectiveness of logic in computer science, 2001
Inductive arguments on programs

Characterization of complexity classes

Inductive definition of type judgments

Expressive Query languages (WITH RECURSIVE) SQL3, IBM DB2, Datalog

Inductive arguments on programs

Characterization of complexity classes

Expressive Query languages (WITH RECURSIVE) SQL3, IBM DB2, Datalog

Inductive definition of type judgments

Reachability properties

Applications of Logic in CS

- Verification
- Database
- Complexity
- Type Theory
- Model Checking
- Knowledge Reasoning

- Inductive arguments on programs
- Expressive Query languages (WITH RECURSIVE) SQL3, IBM DB2, Datalog
- Characterization of complexity classes
- Inductive definition of type judgments
- Reachability properties
- Common knowledge, defined inductively

Applications of Logic in CS

- Verification
- Database
- Complexity
- Type Theory
- Knowledge Reasoning
- Model Checking

- Inductive arguments on programs
- Expressive Query languages (WITH RECURSIVE) SQL3, IBM DB2, Datalog
- Characterization of complexity classes
- Natural numbers
- Inductive definition of type judgments
- Reachability properties
- Common knowledge, defined inductively

What Logic?

- FOL
- SOL

No inductive machinery

Overkill

natural, effective extensions of FOL that allow inductive definitions

Transitive Closure Logic
What Logic?

FOL — SOL

No inductive machinery
What Logic?

FOL: No inductive machinery
SOL: Overkill

natural, effective extensions of FOL that allow inductive definitions: Transitive Closure Logic
natural, effective extensions of *FOL* that allow inductive definitions
What Logic?

- **FOL**
  - No inductive machinery

- **SOL**
  - Overkill

natural, effective extensions of *FOL* that allow inductive definitions

Transitive Closure Logic
Transitive Closure Logic

**Transitive Closure Logic** = $FOL +$ a transitive closure operator.
Transitive Closure Logic $= FOL + \text{ a transitive closure operator.}$

The transitive closure $R^*$ of binary relation $R$ is defined by:

$$R^* = \bigcup R^{(n)}$$

where $R^{(0)} = Id, R^{(n+1)} = R^{(n)} \circ R.$
Transitive Closure Logic = FOL + a transitive closure operator.

The transitive closure $R^*$ of binary relation $R$ is defined by:

$$R^* = \bigcup R^{(n)}$$

where $R^{(0)} = \text{Id}$, $R^{(n+1)} = R^{(n)} \circ R$.

Alternatively,

$$R^* = \text{Id} \cup \bigcap \{ S \mid R \cup S \circ R \subseteq S \}$$

(Least fixed point of the composition operator)
Why Transitive Closure Logic?

- The concept of the transitive closure is truly **basic**.
Why Transitive Closure Logic?

- The concept of the transitive closure is truly **basic**.
  - Being a ‘descendent of’
  - The natural numbers
  - Well-formed formulas
Why Transitive Closure Logic?

- The concept of the transitive closure is truly basic.
  - Being a ‘descendent of’
  - The natural numbers
  - Well-formed formulas
- A minimal extension.
Why Transitive Closure Logic?

- The concept of the transitive closure is truly **basic**.
  - Being a ‘descendent of’
  - The natural numbers
  - Well-formed formulas
- A **minimal** extension.
  - A special case of a least fixed point.
Why Transitive Closure Logic?

- The concept of the transitive closure is truly basic.
  - Being a ‘descendent of’
  - The natural numbers
  - Well-formed formulas
- A minimal extension.
  - A special case of a least fixed point.
- Equivalent to other extensions of FOL, but the most convenient from a proof theoretical perspective.
Why Transitive Closure Logic?

- The concept of the transitive closure is truly basic.
  - Being a ‘descendent of’
  - The natural numbers
  - Well-formed formulas
- A minimal extension.
  - A special case of a least fixed point.
- Equivalent to other extensions of $FOL$, but the most convenient from a proof theoretical perspective.
- Captures inductive principles in a uniform way.
Why Transitive Closure Logic?

- The concept of the transitive closure is truly **basic**.
  - Being a ‘descendent of’
  - The natural numbers
  - Well-formed formulas
- A **minimal** extension.
  - A special case of a least fixed point.
- Equivalent to other extensions of *FOL*, but the most **convenient from a proof theoretical perspective**.
- Captures inductive principles in a **uniform** way.
  - Not parametrized by a set of inductive principles.
The Language
The language $\mathcal{L}_{TC}$ is defined as $\mathcal{L}_{FOL}$, with the additional clause:

- $(RTC_{x,y}\varphi)(s, t)$ is a formula,
  for $\varphi$ a formula, $x, y$ distinct variables, and $s, t$ terms. ($x, y$ become bound in this formula.)
The Language

The language $\mathcal{L}_{TC}$ is defined as $\mathcal{L}_{FOL}$, with the additional clause:

- $(RTC_{x,y}\phi)(s, t)$ is a formula,
  for $\phi$ a formula, $x, y$ distinct variables, and $s, t$ terms.
  ($x, y$ become bound in this formula.)

Allows for:

- Rich testing
- Nested $RTC$
The Intended Meaning of \((RTC_{x,y} \varphi)(s, t)\)

\[
s = t \lor \varphi(s, t) \lor \exists w_1. \varphi(s, w_1) \land \varphi(w_1, t) \\
\lor \exists w_1 \exists w_2. \varphi(s, w_1) \land \varphi(w_1, w_2) \land \varphi(w_2, t) \lor \ldots
\]
The Semantics

The Intended Meaning of $(RTC_{x,y} \varphi)(s, t)$

\[ s = t \lor \varphi(s, t) \lor \exists w_1. \varphi(s, w_1) \land \varphi(w_1, t) \]

\[ \lor \exists w_1 \exists w_2. \varphi(s, w_1) \land \varphi(w_1, w_2) \land \varphi(w_2, t) \lor \ldots \]

Formal Definition

Let $M$ be a structure for $\mathcal{L}_{TC}$ and $\nu$ an assignment in $M$.

$M, \nu \models (RTC_{x,y} \varphi)(s, t)$ iff there exist $a_0, \ldots a_n \in D$ s.t.

$\nu[s] = a_0; \ \nu[t] = a_n; \ M, \nu[x := a_i, y := a_{i+1}] \models \varphi$ for $0 \leq i < n$. 

\[ s \rightarrow \varphi \rightarrow a_0 \rightarrow \varphi \rightarrow a_1 \rightarrow \varphi \rightarrow a_2 \rightarrow \ldots \rightarrow a_{n-1} \rightarrow \varphi \rightarrow a_n \rightarrow t \]
The Intended Meaning of \((RTC_{x,y} \varphi)(s, t)\)

\[ s = t \lor \varphi(s, t) \lor \exists w_1. \varphi(s, w_1) \land \varphi(w_1, t) \]
\[ \lor \exists w_1 \exists w_2. \varphi(s, w_1) \land \varphi(w_1, w_2) \land \varphi(w_2, t) \lor \ldots \]

Formal Definition

Let \(M\) be a structure for \(\mathcal{L}_{TC}\) and \(v\) an assignment in \(M\).

\(M, v \models (RTC_{x,y} \varphi)(s, t)\) iff there exist \(a_0, \ldots a_n \in D\) s.t.
\(v[s] = a_0; v[t] = a_n; M, v[x := a_i, y := a_{i+1}] \models \varphi\) for \(0 \leq i < n\).

\(M, v \models (RTC_{x,y} \varphi)(s, t)\) provided for every \(A \subseteq D\), if \(v(s) \in A\) and \(\forall a, b \in D : (a \in A \land M, v[x := a, y := b] \models \varphi) \rightarrow b \in A\), then \(v(t) \in A\).
The reflexive and the non-reflexive $TC$ operators are equivalent (assuming equality).
Expressive Power

• The reflexive and the non-reflexive $TC$ operators are equivalent (assuming equality).

Theorem [Avron, ’03]

All recursive functions and relations are definable in $\mathcal{L}_{TC}^{\{0,s\}}$
• The reflexive and the non-reflexive $TC$ operators are equivalent (assuming equality).

**Theorem [Avron, ’03]**

All recursive functions and relations are definable in $\mathcal{L}_TC^{\{0,s\}}$ (with pairs)
- The reflexive and the non-reflexive $TC$ operators are equivalent (assuming equality).

**Theorem [Avron, ’03]**

All recursive functions and relations are definable in $\mathcal{L}_{TC}^{\{0,s\}}$ (with pairs)

- $+$ is definable in $\mathcal{L}_{TC}^{\{0,s\}}$ (with pairs) by:
  
  $$x = y + z \iff (RTC_{u,v} v.1 = s(u.1) \land v.2 = s(u.2)) ((0, y), (z, x))$$
The reflexive and the non-reflexive TC operators are equivalent (assuming equality).

**Theorem [Avron, ’03]**

All recursive functions and relations are definable in $\mathcal{L}_{TC}^{\{0,s\}}$ (with pairs)

- $+$ is definable in $\mathcal{L}_{TC}^{\{0,s\}}$ (with pairs) by:

$$x = y + z \iff (RTC_{u,v} \land v.1 = s(u.1) \land v.2 = s(u.2))((0, y), (z, x))$$
• The reflexive and the non-reflexive $TC$ operators are equivalent (assuming equality).

**Theorem [Avron, '03]**

All recursive functions and relations are definable in $\mathcal{L}_{TC}^{\{0,s\}}$ (with pairs)

• $+$ is definable in $\mathcal{L}_{TC}^{\{0,s\}}$ (with pairs) by:

$$x = y + z \iff (RTC_{u,v} v.1 = s(u.1) \land v.2 = s(u.2)) ((0, y), (z, x))$$
Expressive Power

- The reflexive and the non-reflexive $TC$ operators are equivalent (assuming equality).

**Theorem [Avron, '03]**

All recursive functions and relations are definable in $\mathcal{L}_{TC}^{\{0,s\}}$ (with pairs)

- $+$ is definable in $\mathcal{L}_{TC}^{\{0,s\}}$ (with pairs) by:

$x = y + z \iff (RTC_{u,v} v.1 = s(u.1) \land v.2 = s(u.2)) (\langle (0, y), (z, x) \rangle)$
Categorical Characterization of the Natural Numbers

\[ \forall x \ (s(x) \neq 0) \]
\[ \forall x \forall y \ (s(x) = s(y) \rightarrow x = y) \]
\[ \forall x \ (RTC_{w,u} (s(w) = u)) (0, x) \]
Categorical Characterization of the Natural Numbers

\[ \forall x \ (s(x) \neq 0) \]
\[ \forall x \forall y \ (s(x) = s(y) \rightarrow x = y) \]
\[ \forall x \ (RTC_{w,u} (s(w) = u)) (0, x) \]

Corollaries:

- The upward Löwenheim-Skolem theorem fails for TC-logic.
Categorical Characterization of the Natural Numbers

\[ \forall x \ (s(x) \neq 0) \]
\[ \forall x \forall y \ (s(x) = s(y) \rightarrow x = y) \]
\[ \forall x \ (RTC_{w,u} (s(w) = u)) (0, x) \]

Corollaries:

- The upward Löwenheim-Skolem theorem fails for TC-logic.
- TC-logic is not compact.
Categorical Characterization of the Natural Numbers

\[ \forall x \ (s(x) \neq 0) \]
\[ \forall x \forall y \ (s(x) = s(y) \rightarrow x = y) \]
\[ \forall x \ (RTC_{w,u} (s(w) = u)) \ (0, x) \]

Corollaries:

- The upward Löwenheim-Skolem theorem fails for TC-logic.
- TC-logic is not compact.
- TC-logic is inherently incomplete.
Expressive Power

**Categorical Characterization of the Natural Numbers**

\[ \forall x \ (s(x) \neq 0) \]
\[ \forall x \forall y \ (s(x) = s(y) \rightarrow x = y) \]
\[ \forall x \ (RTC_{w,u} (s(w) = u)) (0, x) \]

**Corollaries:**

- The upward Löwenheim-Skolem theorem fails for TC-logic.
- TC-logic is not compact.
- TC-logic is inherently incomplete.
Proof Theory

Infinitary Systems

Completeness

Effectiveness

Finitary Systems
The System $\mathcal{LK}_\rightarrow$ [Gentzen, '34]

\[
\begin{align*}
\frac{\psi, \Gamma \Rightarrow \Delta}{\varphi \land \psi, \Gamma \Rightarrow \Delta} & \quad (\land L_1) \\
\frac{\varphi, \Gamma \Rightarrow \Delta}{\varphi \land \psi, \Gamma \Rightarrow \Delta} & \quad (\land L_2) \\
\frac{\Gamma \Rightarrow \Delta, \varphi \land \psi}{\Gamma \Rightarrow \Delta, \varphi \land \psi} & \quad (\land R)
\end{align*}
\]

\[
\begin{align*}
\frac{\varphi, \Gamma \Rightarrow \Delta \quad \psi, \Gamma \Rightarrow \Delta}{\varphi \lor \psi, \Gamma \Rightarrow \Delta} & \quad (\lor L) \\
\frac{\Gamma \Rightarrow \Delta, \varphi \lor \psi}{\Gamma \Rightarrow \Delta, \varphi \lor \psi} & \quad (\lor R_1) \\
\frac{\Gamma \Rightarrow \Delta, \varphi \lor \psi}{\Gamma \Rightarrow \Delta, \varphi \lor \psi} & \quad (\lor R_2)
\end{align*}
\]

\[
\begin{align*}
\frac{\Gamma \Rightarrow \Delta, \varphi \quad \psi, \Gamma \Rightarrow \Delta}{\varphi \rightarrow \psi, \Gamma \Rightarrow \Delta} & \quad (\rightarrow L) \\
\frac{\Gamma \Rightarrow \Delta, \varphi}{\Gamma \Rightarrow \Delta, \varphi} & \quad (\rightarrow R)
\end{align*}
\]

\[
\begin{align*}
\frac{\Gamma \Rightarrow \Delta, \varphi}{\neg \varphi, \Gamma \Rightarrow \Delta} & \quad (\neg L) \\
\frac{\varphi, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg \varphi} & \quad (\neg R)
\end{align*}
\]

\[
\begin{align*}
\frac{\varphi \left\{ \frac{t}{x} \right\}, \Gamma \Rightarrow \Delta}{\forall x \varphi, \Gamma \Rightarrow \Delta} & \quad (\forall L) \\
\frac{\Gamma \Rightarrow \Delta, \varphi \left\{ \frac{y}{x} \right\}}{\Gamma \Rightarrow \Delta, \forall x \varphi} & \quad (\forall R)^* \\
\frac{\varphi \left\{ \frac{y}{x} \right\}, \Gamma \Rightarrow \Delta}{\exists x \varphi, \Gamma \Rightarrow \Delta} & \quad (\exists L)^* \\
\frac{\Gamma \Rightarrow \Delta, \varphi \left\{ \frac{t}{x} \right\}}{\Gamma \Rightarrow \Delta, \exists x \varphi} & \quad (\exists R)
\end{align*}
\]
The System $\mathcal{LK}_\equiv$ [Gentzen, '34]

\[
\Gamma \Rightarrow \Delta \\
\varphi, \Gamma \Rightarrow \Delta \tag{wkL}
\]

\[
\varphi, \varphi, \Gamma \Rightarrow \Delta \\
\varphi, \Gamma \Rightarrow \Delta \tag{cntL}
\]

\[
\Gamma \Rightarrow \Delta, \varphi \varphi, \Gamma \Rightarrow \Delta \\
\Gamma \Rightarrow \Delta \tag{cut}
\]

\[
\Gamma \Rightarrow \Delta \\
\{ \vec{s} \vec{x} \} \Rightarrow \Delta \{ \vec{s} \vec{x} \} \tag{sub}
\]

\[
\varphi \Rightarrow \varphi \tag{id}
\]

\[
\Gamma \Rightarrow \Delta, s = t \Gamma \Rightarrow \Delta, \varphi \left\{ \frac{s}{x} \right\} \tag{eq}
\]

\[
\Rightarrow t = t \tag{eq}
\]
Finitary Proof System – $RTC_G$

**Reflexivity**

\[ \Gamma \Rightarrow \Delta, (RTC_x, y \varphi) (s, s) \]

**Step**

\[
\begin{align*}
\Gamma &\Rightarrow \Delta, (RTC_x, y \varphi) (s, r) & \Gamma &\Rightarrow \Delta, \varphi \left\{ \frac{r}{x}, \frac{t}{y} \right\} \\
\hline
\Gamma &\Rightarrow \Delta, (TC_x, y \varphi) (s, t)
\end{align*}
\]

**Induction**

\[
\begin{align*}
\Gamma, \psi (x), \varphi (x, y) &\Rightarrow \Delta, \psi \left\{ \frac{y}{x} \right\} \\
\hline
\Gamma, \psi \left\{ \frac{s}{x} \right\}, (RTC_x, y \varphi) (s, t) &\Rightarrow \Delta, \psi \left\{ \frac{t}{x} \right\}
\end{align*}
\]

provided \( x \notin FV (\Gamma \cup \Delta) \) and \( y \notin FV (\Gamma \cup \Delta \cup \{ \psi \}) \).
$\Gamma \Rightarrow \Delta, (RTC_{x,y} \varphi) (s, t)$
$\Gamma \Rightarrow \Delta, (RTC_{y,x} \varphi) (t, s)$

$\Gamma \Rightarrow \Delta, (RTC_{x,y} \varphi) (s, t)$
$\Gamma \Rightarrow \Delta, (RTC_{y,x} \varphi) (t, s)$

$\Gamma \Rightarrow \Delta, (RTC_{x,y} \varphi) (s, t)$
$\Gamma \Rightarrow \Delta, (RTC_{x,y} \varphi) (r, t)$

$\Gamma \Rightarrow \Delta, (RTC_{x,y} \varphi) (s, t)$

$\Gamma, \varphi \Rightarrow \Delta, \psi$

$\Gamma, (RTC_{x,y} \varphi) (s, t) \Rightarrow \Delta, (RTC_{x,y} \psi) (s, t)$

$\Gamma, (RTC_{x,y} \varphi) (s, t), \Gamma \Rightarrow \Delta$

$(RTC_{u,v} (RTC_{x,y} \varphi) (u, v)) (s, t), \Gamma \Rightarrow \Delta$

$\Gamma \Rightarrow \Delta, (RTC_{x,y} \varphi) (s, t)$

$\Gamma \Rightarrow \Delta, s = t, \exists z ((RTC_{x,y} \varphi) (s, z) \land \varphi \left\{ \frac{z}{x}, \frac{t}{y} \right\})$
**TC for Arithmetics**

\(RTC_G + A\) is obtained from \(RTC_G\) by the addition of the standard axioms for successor and addition, and the axiom characterizing the natural numbers in TC-logic.
TC for Arithmetics

$\text{RTC}_G + A$ is obtained from $\text{RTC}_G$ by the addition of the standard axioms for successor and addition, and the axiom characterizing the natural numbers in TC-logic.

Theorem

$\text{RTC}_G + A$ is equivalent to the sequent calculi of $\text{PA}$, i.e. there is a provability preserving translation algorithm between them.
Arithmetics in \(RTC_G\)

**TC for Arithmetics**

\(RTC_G + A\) is obtained from \(RTC_G\) by the addition of the standard axioms for successor and addition, and the axiom characterizing the natural numbers in TC-logic.

**Theorem**

\(RTC_G + A\) is equivalent to thesequent calculi of \(PA\), i.e. there is a provability preserving translation algorithm between them.

**Corollary**

The ordinal number of the \(RTC_G + A\) is \(\varepsilon_0\).
A $\sigma$-Henkin structure is a triple $M = \langle D, I, D' \rangle$ (frame), s.t.:

1. $\langle D, I \rangle$ is a FO structure for $\sigma$
2. $D' \subseteq P(D)$ is closed under parametric definability.
A **σ-Henkin structure** is a triple $M = \langle D, I, D' \rangle$ (frame), s.t.:

1. $\langle D, I \rangle$ is a FO structure for $\sigma$
2. $D' \subseteq P(D)$ is closed under parametric definability.

For every $A \in D'$, if $v(s) \in A$ and $\forall a, b \in D : (a \in A \land M, v[x := a, y := b] \models \varphi) \rightarrow b \in A$, then $v(t) \in A$. 

Completeness Theorem

$T \vdash RTC_{x,y} \varphi \iff T \models H \varphi$. 

---

**Henkin Semantics**
A σ-Henkin structure is a triple $M = \langle D, I, D' \rangle$ (frame), s.t.:

1. $\langle D, I \rangle$ is a FO structure for $\sigma$
2. $D' \subseteq P(D)$ is closed under parametric definability.

$M, v \models (RTC_{x,y}\varphi)(s, t)$ provided for every $A \in D'$, if $v(s) \in A$ and $\forall a, b \in D : (a \in A \land M, v [x := a, y := b] \models \varphi) \rightarrow b \in A$, then $v(t) \in A$.

Completeness Theorem
$T \vdash_{RTC_G} \varphi \iff T \models_H \varphi.$
So Far

standard validity

Henkin validity
So Far

standard validity \rightarrow \text{Henkin validity}

RTC \rightarrow RTC_G
Infinitary Systems

Infinitary rules: finite proofs

Finite rules: infinite proofs

Non-effective: can be effective?
Infinitary Systems

Infinitary?

width

infinite rules
finite proofs
finite rules
infinite proofs

non-effective
can be effective?
Infinitary Systems

- Width
  - Infinite rules
  - Finite proofs

Infinitary?
Infinitary Systems

- **width**
  - infinite rules
  - finite proofs

- **height**

- **Infinitary ?**
Infinitary Systems

- **Infinitary ?**
  - **width**
    - infinite rules
    - finite proofs
  - **height**
    - finite rules
    - infinite proofs
Infinitary Systems

- **Width**
  - Infinite rules
  - Finite proofs
  - Non-effective

- **Height**
  - Finite rules
  - Infinite proofs

**Infinitary?**

Can be effective?
Infinitary Systems

- **Width**: Infinite rules, finite proofs → Non-effective
- **Height**: Finite rules, infinite proofs → Can be effective?
Infinite Descent-Style Proof System

Infinite height, not width
Proofs can be infinite, non-well-founded trees, provided that every infinite path admits some *infinite descent*. The descent is witnessed by tracing terms/formulas corresponding to elements of a well-founded set. This *global trace condition* is decidable using Büchi automata. Systems of *implicit induction*. **Infinite height, not width**
Infinitary Proof System – $\RTC^\omega_G$

**Reflexivity**

\[ \Gamma \Rightarrow \Delta, (RTC_{x,y} \varphi) (s, s) \]

**Step**

\[ \Gamma \Rightarrow \Delta, (RTC_{x,y} \varphi) (s, r) \quad \Gamma \Rightarrow \Delta, \varphi \left\{ \frac{r}{x}, \frac{t}{y} \right\} \]

\[ \Gamma \Rightarrow \Delta, (TC_{x,y} \varphi) (s, t) \]

**Case-split**

\[ \Gamma, s = t \Rightarrow \Delta \quad \Gamma, (RTC_{x,y} \varphi)(s, z), \varphi \left\{ \frac{z}{x}, \frac{t}{y} \right\} \Rightarrow \Delta \]

\[ \Gamma, (RTC_{x,y} \varphi)(s, t) \Rightarrow \Delta \]

provided $z$ is fresh.
Infinitary Proof System – $\text{RTC}_G^\omega$

**Reflexivity**

$$\Gamma \Rightarrow \Delta, (\text{RTC}_{x,y} \varphi)(s, s)$$

**Step**

$$\Gamma \Rightarrow \Delta, (\text{RTC}_{x,y} \varphi)(s, r) \quad \Gamma \Rightarrow \Delta, \varphi \left\{ \frac{r}{x}, \frac{t}{y} \right\}$$

$$\Gamma \Rightarrow \Delta, (\text{TC}_{x,y} \varphi)(s, t)$$

**Case-split**

$$\Gamma, s = t \Rightarrow \Delta \quad \Gamma, (\text{RTC}_{x,y} \varphi)(s, z), \varphi \left\{ \frac{z}{x}, \frac{t}{y} \right\} \Rightarrow \Delta$$

$$\Gamma, (\text{RTC}_{x,y} \varphi)(s, t) \Rightarrow \Delta$$

provided $z$ is fresh.
Soundness and Completeness

**Completeness Theorem**

\[ T \vdash^{cf}_{RTC_G^\omega} \varphi \iff T \models \varphi. \]
Soundness and Completeness

**Completeness Theorem**

\[ T \vdash^{cf}_{RTC^G} \varphi \iff T \models \varphi. \]

Global soundness via an infinite descent proof-by-contradiction:
Completeness Theorem

\[ T \vdash^{cf}_{RTC^G}\varphi \iff T \models \varphi. \]

Global soundness via an infinite descent proof-by-contradiction:

- Assume the conclusion of the proof is invalid
Soundness and Completeness

Completeness Theorem

\[ T \models^{cf}_{\text{RTC}_G^\omega} \varphi \iff T \models \varphi. \]

Global soundness via an infinite descent proof-by-contradiction:

- Assume the conclusion of the proof is invalid
- Local soundness entails an infinite sequence of counter models
Completeness Theorem

\[ T \vdash_{RTC_G}^c \varphi \iff T \models \varphi. \]

**Global soundness** via an infinite descent proof-by-contradiction:

- Assume the conclusion of the proof is invalid
- Local soundness entails an infinite sequence of counter models
  - Mapped to the minimal length for witnessing the transitive closure trace.
Soundness and Completeness

Completeness Theorem

\[ T \vdash_{\text{RTC}}^{\text{cf}} \varphi \iff \models_T \varphi. \]

Global soundness via an infinite descent proof-by-contradiction:

- Assume the conclusion of the proof is invalid
- Local soundness entails an infinite sequence of counter models
  - Mapped to the minimal length for witnessing the transitive closure trace.
- Global trace condition entails the chain is infinitely descending...
Soundness and Completeness

Completeness Theorem

\[ T \vdash^\text{cf}_{\text{RTC}_G^\omega} \varphi \iff T \models \varphi. \]

Global soundness via an infinite descent proof-by-contradiction:

- Assume the conclusion of the proof is invalid
- Local soundness entails an infinite sequence of counter models
  - Mapped to the minimal length for witnessing the transitive closure trace.
- Global trace condition entails the chain is infinitely descending
  - But the numbers are well-founded . . . contradiction!
So Far

standard validity

Henkin validity

$RTC_G$
So Far

standard validity

Henkin validity

(cut-free) RTC\(_G^\omega\)

RTC\(_G\)
Proof Theory

Infinitary Systems

Completeness

Effectiveness

Finitary Systems
An effective subsystem can be obtained by considering only the regular infinite proofs. Regular proofs = represented as finite, possibly cyclic, graphs.
An effective subsystem can be obtained by considering only the regular infinite proofs.

Regular proofs $=$ represented as finite, possibly cyclic, graphs.
Implicit Induction Subsumes Explicit Induction

\[ \Gamma, \psi \left\{ \frac{v}{x} \right\}, (RTC_{x,y} \varphi)(v, w) \Rightarrow \Delta, \psi \left\{ \frac{w}{x} \right\} \]  
(Subst)

\[ \Gamma, \psi \left\{ \frac{v}{x} \right\}, (RTC_{x,y} \varphi)(v, z) \Rightarrow \Delta, \psi \left\{ \frac{z}{x} \right\} \]  
(Subst)

\[ \Gamma, \psi \left\{ \frac{v}{x} \right\}, (RTC_{x,y} \varphi)(v, z), \varphi \left\{ \frac{z}{x}, \frac{w}{y} \right\} \Rightarrow \Delta, \psi \left\{ \frac{w}{x} \right\} \]  
(Cut)

\[ \psi \left\{ \frac{v}{x} \right\}, v = w \Rightarrow \psi \left\{ \frac{w}{x} \right\} \]  
(Eq)

\[ \Gamma, \psi \left\{ \frac{s}{x} \right\}, (RTC_{x,y} \varphi)(s, t) \Rightarrow \Delta, \psi \left\{ \frac{t}{x} \right\} \]  
(Subst)

Normal Cyclic Proofs = non-overlapping cyclic proofs.
Every infinite path (from conclusion to premise) is eventually followed by a trace of \( RTC \)-formulas (on the left-hand side) which progresses (via case-split) infinitely often.
Implicit Induction Subsumes Explicit Induction

Every infinite path (from conclusion to premise) is eventually followed by a trace of \( RTC \)-formulas (on the left-hand side) which progresses (via case-split) infinitely often.

• Normal Cyclic Proofs = non-overlapping cyclic proofs.
Cyclic Proof vs. Explicit Induction

Induction invariant

- Complex induction schemes naturally represented by nested and overlapping cycles.
- Every sequent provable using the explicit induction rule is also derivable using cyclic proof.
Cyclic Proof vs. Explicit Induction

- **Induction invariant**
- **Explicit induction** requires it *a priori*
  - Major challenge for automatic proof search

Every sequent provable using the explicit induction rule is also derivable using cyclic proof.
Cyclic Proof vs. Explicit Induction

Induction invariant

Explicit induction requires it \textit{a priori}

Major challenge for automatic proof search

Cyclic proof enables its `discovery`

More exploratory approach to proof search
Cyclic Proof vs. Explicit Induction

- Explicit induction requires it a priori
  - Major challenge for automatic proof search
- Cyclic proof enables its ‘discovery’
  - More exploratory approach to proof search

- Complex induction schemes naturally represented by nested and overlapping cycles.
Cyclic Proof vs. Explicit Induction

- **Induction invariant**
  - Explicit induction requires it *a priori*
  - Cyclic proof enables its ‘discovery’

- Major challenge for automatic proof search
- More exploratory approach to proof search

- Complex induction schemes naturally represented by nested and overlapping cycles.
- Every sequent provable using the explicit induction rule is also derivable using cyclic proof.
So Far

standard validity

(cut-free) $\text{RTC}_G^\omega$

Henkin validity

$\text{RTC}_G$
So Far

standard validity

Henkin validity

(cut-free)

$\text{RTC}_G^\omega$

$\text{RTC}_G$

$\text{CRTC}_G^\omega$
So Far

standard validity

Henkin validity

(cut-free) RTC\(^\omega\)\(_G\)

RTC\(^\omega\)\(_G\)

CRTC\(^\omega\)\(_G\)

NCRTC\(^\omega\)\(_G\)
• For arithmetics, the explicit and cyclic systems are equivalent.
Is the Cyclic System Stronger?

- For arithmetics, the explicit and cyclic systems are equivalent.

- In general, the question of the (in)equivalence between the systems remains open.
Is the Cyclic System Stronger?

• For arithmetics, the explicit and cyclic systems are equivalent.

• In general, the question of the (in)equivalence between the systems remains open.

• In systems for FOL with inductive definition, the equivalence was refuted when both systems have the same set of inductive definitions. [Berardi, Tatsuta, 2017]
• For arithmetics, the explicit and cyclic systems are equivalent.

• In general, the question of the (in)equivalence between the systems remains open.

• In systems for FOL with inductive definition, the equivalence was refuted when both systems have the same set of inductive definitions. [Berardi, Tatsuta, 2017]

• In the TC framework all inductive definitions at once.
So Far

standard validity

(cut-free) 
\( \text{RTC}_G^{\omega} \)

Henkin validity

\( \text{RTC}_G \)

\( \text{CRTC}_G^{\omega} \)

\( \text{NCRTC}_G^{\omega} \)
So Far

standard validity

(cut-free)

\( RTC^\omega_G \)

Henkin validity

\( RTC^\omega_G \)

\( G \)

\( CRTC^\omega_G \)

\( G \)

\( RTC^\omega_G + A \)

\( RTC^\omega_G \)

\( NCRTC^\omega_G \)

\( RTC^\omega_G + A \)

\( G \)
Future (and Current) Work

- Resolving the open question of the (in)equivalence of $RTC_G$ and $CRTC_G^\omega$.
- Implementing $CRTC_G^\omega$ and investigating the practicalities of TC-logic to support automated inductive reasoning.
- Using the uniformity of TC-logic to better study the relationship between implicit and explicit induction.
  - Cuts required in each system
  - Relative complexity of proofs
- Incorporating coinductive reasoning into the formal system.
Summary

standard validity

(cut-free) \( \text{RTC}_G^\omega \)

Henkin validity

\( \text{RTC}_G \)

\( \text{CRTC}_G^\omega \)

\( \text{NCRTC}_G^\omega \)

\( \text{CRTC}_G^\omega + A \)

\( \text{RTC}_G + A \)

Thank you