Transitive Closure Logic

Infinitary and Cyclic Proof Systems

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

PARIS Workshop @ FLoC, Sunday 8th July 2018, Oxford, UK

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

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

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

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

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

The formal semantics:

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

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

The formal semantics:

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

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

The formal semantics:

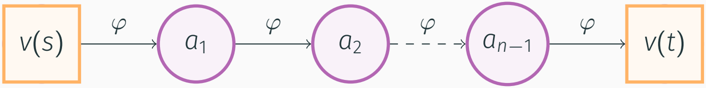
- M is a (standard) first-order model with domain D
- v is a valuation of terms in M:

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

The formal semantics:

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

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



Why ‘Transitive Closure’ logic?

Why ‘Transitive Closure’ logic?

- Consider the binary relation induced by φ (wrt. x and y):

$$[\![\varphi(x,y)]\!]_{M,v} = \{ \ (a,b) \ \mid \ M, v[x := a, y := b] \models \varphi \ \}$$

Why 'Transitive Closure' logic?

- Consider the binary relation induced by φ (wrt. x and y):

$$[\![\varphi(x, y)]\!]_{M,v} = \{ (a, b) \mid M, v[x := a, y := b] \models \varphi \}$$

- $(RTC_{x,y}\, \varphi)$ 'denotes' the reflexive, transitive closure of φ:

$$M, v \models (RTC_{x,y}\, \varphi)(s, t) \Leftrightarrow (v(s), v(t)) \in ([\![\varphi(x, y)]\!]_{M,v})^*$$

Why Transitive Closure logic?

- It is a minimal extension of FOL
- It has an intuitive, easy-to-understand semantics
- It turns out to be surprisingly expressive

Why Transitive Closure logic?

- It is a minimal extension of FOL
- It has an intuitive, easy-to-understand semantics
- It turns out to be surprisingly expressive

Theorem (Avron '03)

All finitely inductively defined relations are definable in* TC.†

A. Avron, *Transitive Closure and the Mechanization of Mathematics*, 2003.

---

*as defined in: S. Feferman, *Finitary Inductively Presented Logics*, 1989

†with signatures containing a pairing function

Example: Arithmetic

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

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

Example: Arithmetic

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

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

Example: Arithmetic

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

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

$$\text{“}x = y + z\text{”} \equiv$$

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

Example: Arithmetic

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

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

$$\text{“}x = y + z\text{”} \equiv$$

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

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

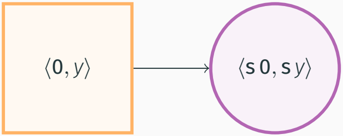
Example: Arithmetic

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

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

$$\text{“}x = y + z\text{”} \equiv$$

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



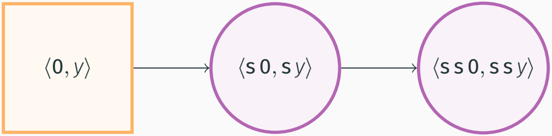
Example: Arithmetic

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

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

$$\text{“}x = y + z\text{”} \equiv$$

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



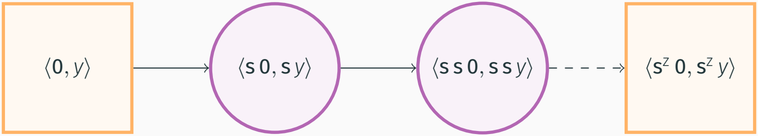
Example: Arithmetic

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

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

$$\text{“}x = y + z\text{”} \equiv$$

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



Example: Arithmetic

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

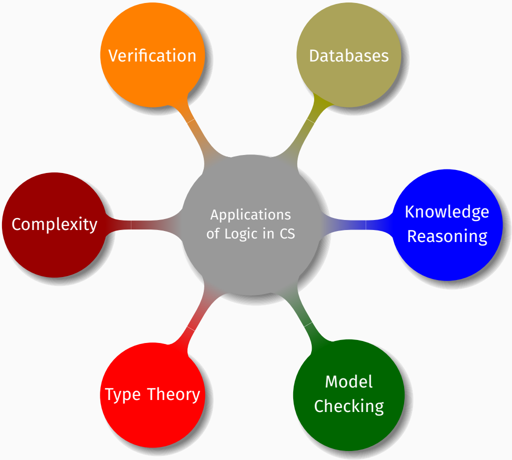
$$\mathsf{Nat}(x) \equiv (RTC_{v,w}\, \mathsf{s}\, v = w)(\mathsf{0}, x)$$

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

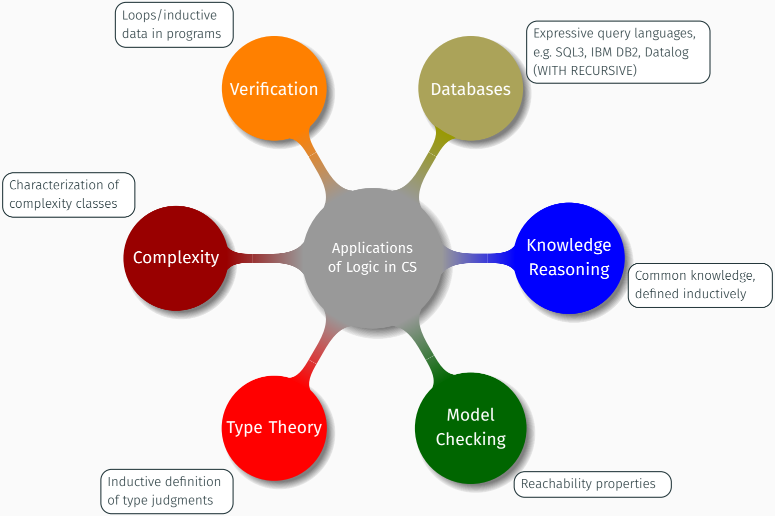
$$(RTC_{v,w}\, \exists n_1, n_2 \,.\, v = \langle n_1, n_2 \rangle \wedge w = \langle \mathsf{s}\, n_1, \mathsf{s}\, n_2 \rangle)(\langle \mathsf{0}, y \rangle, \langle z, x \rangle)$$

- The following axioms categorically characterise the natural numbers in TC:

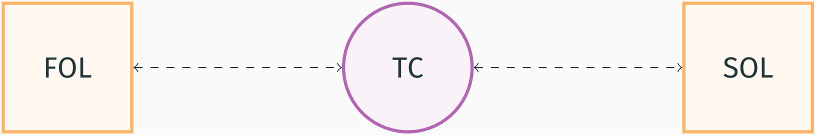
$$\forall x \,.\, \mathsf{s}\, x \neq \mathsf{0}$$
$$\forall x, y \,.\, \mathsf{s}\,(x) = \mathsf{s}\,(y) \rightarrow x = y$$
$$\forall x \,.\, \mathsf{Nat}(x)$$

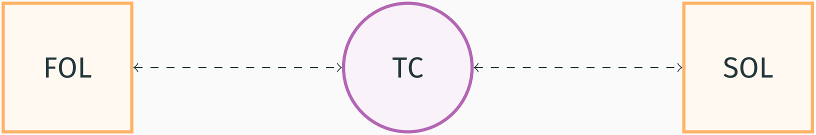


J. Halpern Et Al, *On the Unusual Effectiveness of Logic in Computer Science*, 2001



J. Halpern Et Al, *On the Unusual Effectiveness of Logic in Computer Science*, 2001

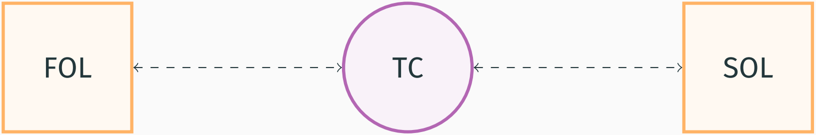

FOL
TC
SOL



"Everything should be made as simple as possible but not simpler"

—Albert Einsten

Weak **SOL**
ω-logic
Cardinality logic
FOL + Henkin Quantifiers



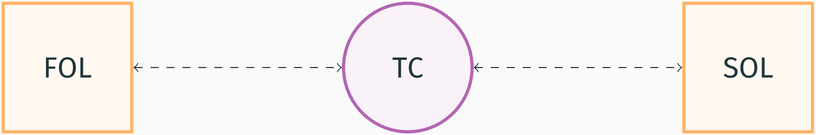
"Everything should be made as simple as possible but not simpler"
—Albert Einsten

Weak **SOL**

ω-logic

Cardinality logic

FOL + Henkin Quantifiers



FOL
+ ML Ind.
Defs

FOMμ

"Everything should be made as simple as possible but not simpler"

—Albert Einsten

The transitive closure

$$R^+ = \bigcup_{i \geq 0} R_i, \quad \text{where } R_0 = R$$
$$R_{i+1} = R_i \circ R \quad (i \geq 0)$$

is a particular kind of fixed point:

$$R^+ = \mu X.\Psi_R(X)$$

where, for binary relations R and S, we define

$$\Psi_R(S) = R \cup (R \circ S)$$

FOL + Martin-Löf inductive definitions:

- For each predicate symbol $P_1, \ldots, P_n$, we give a set of productions of the form:

$$\frac{Q_1(\vec{s_1}) \quad \ldots \quad Q_n(\vec{s_n})}{P_i(\vec{t})}$$

FOL + Martin-Löf inductive definitions:

- For each predicate symbol $P_1, \ldots, P_n$, we give a set of productions of the form:

$$\frac{Q_1(\vec{s_1}) \quad \ldots \quad Q_n(\vec{s_n})}{P_i(\vec{t})}$$

- The productions induce a monotone operator on the domain of predicate interpretations $\mathcal{X}$:

$$\mathcal{X} : Pred \to \wp(\bigcup_{k \geq 0} \mathcal{D}^k)$$

FOL + Martin-Löf inductive definitions:

- For each predicate symbol $P_1, \ldots, P_n$, we give a set of productions of the form:

$$\frac{Q_1(\vec{s_1}) \quad \ldots \quad Q_n(\vec{s_n})}{P_i(\vec{t})}$$

- The productions induce a monotone operator on the domain of predicate interpretations $\mathcal{X}$:

$$\mathcal{X} : Pred \to \wp(\bigcup_{k \geq 0} \mathcal{D}^k)$$

- The semantics of the logic uses the least fixed point

FOL + Martin-Löf inductive definitions:

- For each predicate symbol $P_1, \ldots, P_n$, we give a set of productions of the form:

$$\frac{Q_1(\vec{s_1}) \quad \ldots \quad Q_n(\vec{s_n})}{P_i(\vec{t})}$$

TC has all possible inductive definitions 'available' using only a finite signature

onotone operator on the

etations $\mathcal{X}$:

$$\rightarrow \wp(\bigcup_{k \geq 0} \mathcal{D}^k)$$

- The semantics of the logic uses the least fixed point

FOL + Martin-Löf inductive definitions:

- For each predicate symbol $P_1, \ldots, P_n$, we give a set of productions of the form:

$$\frac{Q_1(\vec{s_1}) \quad \ldots \quad Q_n(\vec{s_n})}{P_i(\vec{t})}$$

- The productions induce a monotone operator on the domain of predicate interpretations $\mathcal{X}$:

FOL$_{\text{ID}}$ productions only allow for Horn clauses

- The semantics of the logic uses the least fixed point

What about the proof theory?

What about the proof theory?

EFFECTIVE

COMPLETE

What about the proof theory?

Effective	Complete

What about the proof theory?

Effective	Complete
Finitary RTC_G	

What about the proof theory?

Effective

Finitary
RTC_G

Complete

Infinitary
RTC_G^ω

What about the proof theory?

Effective

Finitary
RTC_G

Cyclic
CRTC^{ω}_G

Complete

Infinitary
RTC^{ω}_G

What about the proof theory?

Effective

Finitary

RTC_G

↓

Cyclic

CRTC^{ω}_G

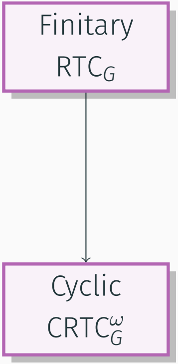
Complete

Infinitary

RTC^{ω}_G

What about the proof theory?

Effective Henkin-Complete



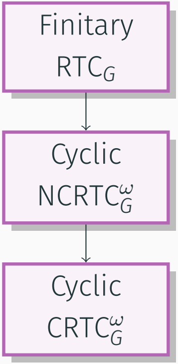
Complete

Infinitary

RTC^{ω}_G

What about the proof theory?

EFFECTIVE HENKIN-COMPLETE



COMPLETE

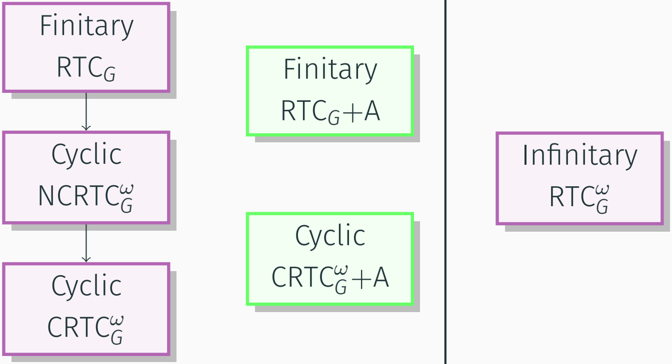
Infinitary

RTC_G^ω

What about the proof theory?

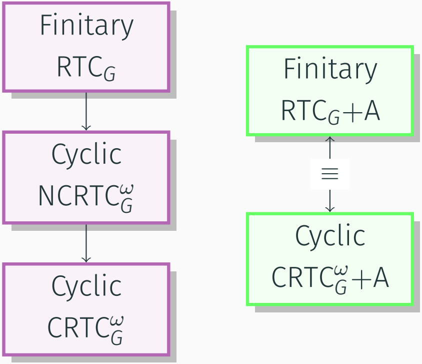
Effective Henkin-Complete

Complete



What about the proof theory?

Effective Henkin-Complete



Complete

RTC_G: A Finitary Proof System with ‘Explicit’ Induction

We add the following rules to Gentzen’s sequent calculus for CL with substitution and equality:

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

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

RTC$_G$ ‘captures’ TC:

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

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

$$\frac{\Gamma, \varphi[s/x] \vdash \Delta}{\Gamma, (RTC_{x,y}\,\varphi)(s,t) \vdash \Delta, s = t} \qquad \frac{\Gamma, (RTC_{x,y}\,\varphi)(s,t) \vdash \Delta}{\Gamma, (RTC_{v,w}\,(RTC_{x,y}\,\varphi)(v,w))(s,t) \vdash \Delta}$$

$$\frac{\Gamma \vdash \Delta, (RTC_{x,y}\,\varphi)(s,t)}{\Gamma \vdash \Delta, s = t, \exists z\,.\,(RTC_{x,y}\,\varphi)(s,z) \wedge \varphi[z/x, t/y]}$$

RTC_G is complete for the following Henkin-style semantics:

RTC_G is complete for the following Henkin-style semantics:

- A **TC** Henkin-frame H is a triple $\langle D, I, \mathcal{D} \rangle$
  - $\langle D, I \rangle$ is a first-order structure
  - $\mathcal{D} \subseteq \wp(D)$ is its set of admissible subsets

RTC_G is complete for the following Henkin-style semantics:

- A **TC** Henkin-frame H is a triple $\langle D, I, \mathcal{D} \rangle$
    - $\langle D, I \rangle$ is a first-order structure
    - $\mathcal{D} \subseteq \wp(D)$ is its set of admissible subsets
- *RTC* formulas are interpreted wrt. frames as follows:

$$H, v \models_{\mathcal{H}} (RTC_{x,y}\, \varphi)(s,t) \Leftrightarrow$$
$$\text{for all } A \in \mathcal{D}, \text{ if } v(s) \in A \text{ and}$$
$$\forall a, b \in D \,.\, (a \in A \wedge H, v[x := a, y := b] \models \varphi) \rightarrow b \in A$$
$$\text{then } v(t) \in A$$

RTC$_G$ is complete for the following Henkin-style semantics:

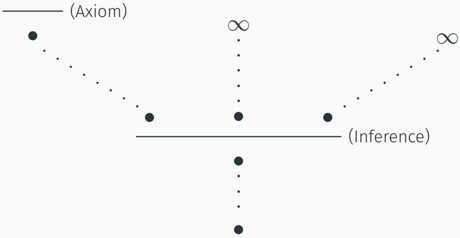
- A **TC** Henkin-frame H is a triple $\langle D, I, \mathcal{D} \rangle$
  - $\langle D, I \rangle$ is a first-order structure
  - $\mathcal{D} \subseteq \wp(D)$ is its set of admissible subsets
- *RTC* formulas are interpreted wrt. frames as follows:

$$H, v \models_{\mathcal{H}} (RTC_{x,y}\, \varphi)(s, t) \Leftrightarrow$$
$$\text{for all } A \in \mathcal{D}, \text{ if } v(s) \in A \text{ and}$$
$$\forall a, b \in D\,.\,(a \in A \wedge H, v[x := a, y := b] \models \varphi) \rightarrow b \in A$$
$$\text{then } v(t) \in A$$

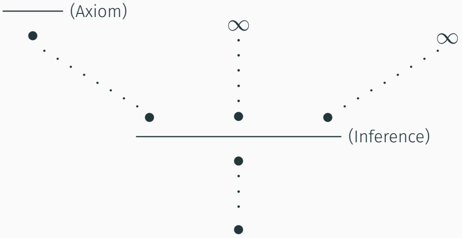
- A **TC** Henkin structure is a **TC** Henkin-frame closed under parametric definability, i.e.

$$\{a \in D \mid H, v[x := a] \models \varphi\} \in \mathcal{D} \text{ for all } \varphi, v, \text{ and } H$$

In non-well-founded proof theory we allow infinite height derivations:

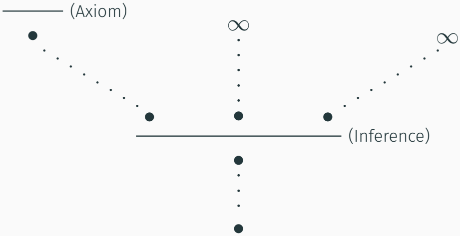


In non-well-founded proof theory we allow infinite height derivations:



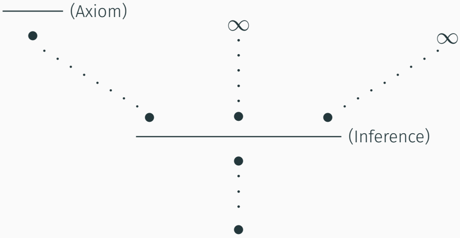
- We only accept proofs for which every path admits some infinite descent

In non-well-founded proof theory we allow infinite height derivations:



- We only accept proofs for which every path admits some infinite descent
- This is witnessed by tracing terms/formulas corresponding to elements of a well-founded set

In non-well-founded proof theory we allow infinite height derivations:



- We only accept proofs for which every path admits some infinite descent
- This is witnessed by tracing terms/formulas corresponding to elements of a well-founded set
- This global trace condition is an ω-regular property (i.e. decidable using Büchi automata)

RTC_G^{ω}: An Infinitary Proof System with 'Implicit' Induction

We simply replace the explicit induction rule of RTC_G with:

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

RTC_G^ω: An Infinitary Proof System with ‘Implicit’ Induction

We simply replace the explicit induction rule of RTC_G with:

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

We trace formulas $(RTC_{x,y}\,\varphi)(s,t)$ in the antecedent of sequents

RTC_G^ω: An Infinitary Proof System with 'Implicit' Induction

We simply replace the explicit induction rule of RTC_G with:

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

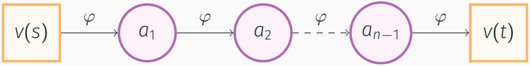
We trace formulas $(RTC_{x,y}\,\varphi)(s,t)$ in the antecedent of sequents

The trace progresses when it traverses the principal formula of a case-split rule.

Soundness of RTC_G^ω

- Define a measure function for *RTC*-formulas:

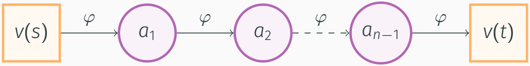
$$\delta_{(RTC_{x,y}\,\varphi)(s,t)}(M, v) = \begin{cases} \text{minimal no. of } \varphi\text{-steps} \\ \text{from } v(s) \text{ to } v(t) \text{ in } M \end{cases}$$



Soundness of RTC_G^ω

- Define a measure function for *RTC*-formulas:

$$\delta_{(RTC_{x,y}\,\varphi)(s,t)}(M, v) = \begin{cases}\text{minimal no. of } \varphi\text{-steps} \\ \text{from } v(s) \text{ to } v(t) \text{ in } M\end{cases}$$



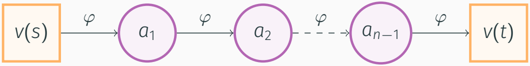
- The proof rules have the following property:

$$\frac{\Gamma_1 \vdash \Delta_1 \quad \dots \quad \Gamma_n \vdash \Delta_n}{\Gamma \vdash \Delta}$$

Soundness of RTC_G^{ω}

- Define a measure function for *RTC*-formulas:

$$\delta_{(RTC_{x,y}\,\varphi)(s,t)}(M,v) = \begin{cases} \text{minimal no. of } \varphi\text{-steps} \\ \text{from } v(s) \text{ to } v(t) \text{ in } M \end{cases}$$



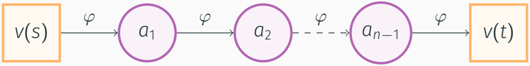
- The proof rules have the following property:

$$\frac{\Gamma_1 \vdash \Delta_1 \quad \dots \quad (M',v') \not\models \Gamma_i \vdash \Delta_i \quad \dots \quad \Gamma_n \vdash \Delta_n}{(M,v) \not\models \Gamma \vdash \Delta}$$

Soundness of RTC_G^{ω}

- Define a measure function for *RTC*-formulas:

$$\delta_{(RTC_{x,y}\,\varphi)(s,t)}(M,v) = \begin{cases}\text{minimal no. of } \varphi\text{-steps} \\ \text{from } v(s) \text{ to } v(t) \text{ in } M\end{cases}$$



- The proof rules have the following property:

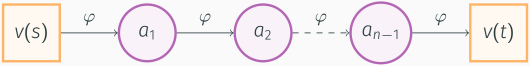
$$\frac{\ldots \quad (M',v') \not\models \Gamma_i, (RTC_{v,w}\,\varphi')(r,u) \vdash \Delta_i \quad \ldots}{(M,v) \not\models \Gamma, (RTC_{x,y}\,\varphi)(s,t) \vdash \Delta}$$

$$\delta_{(RTC_{v,w}\,\varphi')(r,u)}(M',v') \leq \delta_{(RTC_{x,y}\,\varphi)(s,t)}(M,v)$$

Soundness of RTC_G^ω

- Define a measure function for *RTC*-formulas:

$$\delta_{(RTC_{x,y}\,\varphi)(s,t)}(M,v) = \begin{cases} \text{minimal no. of } \varphi\text{-steps} \\ \text{from } v(s) \text{ to } v(t) \text{ in } M \end{cases}$$



- The proof rules have the following property:

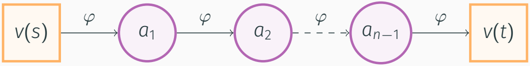
$$\frac{\Gamma, s = t \vdash \Delta \quad (M', v') \not\models \Gamma, (RTC_{x,y}\,\varphi)(s,z), \varphi[z/x, t/y] \vdash \Delta}{(M,v) \not\models \Gamma, (RTC_{x,y}\,\varphi)(s,t) \vdash \Delta}$$

$$\delta_{(RTC_{v,w}\,\varphi')(r,u)}(M',v') < \delta_{(RTC_{x,y}\,\varphi)(s,t)}(M,v)$$

Soundness of RTC_G^ω

- Define a measure function for *RTC*-formulas:

$$\delta_{(RTC_{x,y}\,\varphi)(s,t)}(M,v) = \begin{cases} \text{minimal no. of } \varphi\text{-steps} \\ \text{from } v(s) \text{ to } v(t) \text{ in } M \end{cases}$$



- The proof rules have the following property:

$$\frac{\Gamma, s = t \vdash \Delta \quad (M', v') \not\models \Gamma, (RTC_{x,y}\,\varphi)(s,z), \varphi[z/x, t/y] \vdash \Delta}{(M,v) \not\models \Gamma, (RTC_{x,y}\,\varphi)(s,t) \vdash \Delta}$$

$$\delta_{(RTC_{v,w}\,\varphi')(r,u)}(M',v') < \delta_{(RTC_{x,y}\,\varphi)(s,t)}(M,v)$$

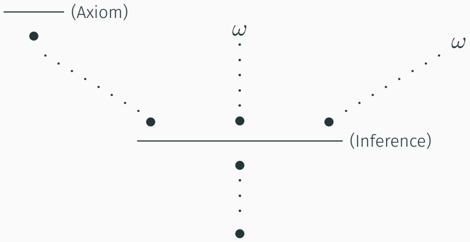
- Global trace condition $\Rightarrow n_1 > n_2 > n_3 > \ldots$

Cut-free Completeness of RTC_G^ω

Obtained using a variation of the standard technique:

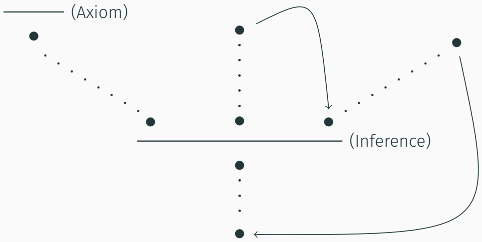
1. Construct an infinite (cut-free) pre-proof via an exhaustive search tree
2. If not a valid proof, then it is possible to construct a counter-model
3. Thus search tree gives a valid proof for every valid sequent

CRTC_G^ω: A Cyclic Subsystem



- Restricting to all and only regular infinite pre-proofs gives an effective system

CRTC_G^ω: A Cyclic Subsystem



- Restricting to all and only regular infinite pre-proofs gives an effective system
- Regular pre-proofs can be represented as finite, possibly cyclic graphs

Implicit induction subsumes explicit induction

$$
\dfrac{
\dfrac{
\dfrac{}{\Gamma, \psi[v/x] \vdash \Delta, \psi[v/x]}\,(\mathsf{Ax})
}{
\Gamma, \psi[v/x], v = w \vdash \Delta, \psi[w/x]
}\,({=}\mathsf{L})
\qquad
\begin{array}{c}
\dfrac{
\dfrac{\Gamma, \psi[v/x], (RTC_{x,y}\,\varphi)(v, w) \vdash \Delta, \psi[w/x]}{\Gamma, \psi[v/x], (RTC_{x,y}\,\varphi)(v, z) \vdash \Delta, \psi[z/x]}\,(\mathsf{Subst})
\quad
\dfrac{\begin{array}{c}\vdots \\ \Gamma, \psi, \varphi \vdash \Delta, \psi[y/x]\end{array}}{\Gamma, \psi[z/x], \varphi[z/x, w/y] \vdash \Delta, \psi[w/x]}\,(\mathsf{Subst})
}{
\Gamma, \psi[v/x], (RTC_{x,y}\,\varphi)(v, z), \varphi[z/x, w/y] \vdash \Delta, \psi[w/x]
}\,(\mathsf{Cut}) \\
\vdots
\end{array}
}{
\dfrac{\Gamma, \psi[v/x], (RTC_{x,y}\,\varphi)(v, w) \vdash \Delta, \psi[w/x]}{\begin{array}{c}\Gamma, \psi[s/x], (RTC_{x,y}\,\varphi)(s, t) \vdash \Delta, \psi[t/x] \\ \vdots\end{array}}\,(\mathsf{Subst})
}\,(\mathsf{case\text{-}split})
$$

Implicit induction subsumes explicit induction

$$\dfrac{\Gamma, \psi[v/x], (RTC_{x,y}\,\varphi)(v,w) \vdash \Delta, \psi[w/x]}{\Gamma, \psi[v/x], (RTC_{x,y}\,\varphi)(v,z) \vdash \Delta, \psi[z/x]}\ (\text{Subst}) \qquad \dfrac{\begin{array}{c}\vdots \\ \Gamma, \psi, \varphi \vdash \Delta, \psi[y/x]\end{array}}{\Gamma, \psi[z/x], \varphi[z/x, w/y] \vdash \Delta, \psi[w/x]}\ (\text{Subst})$$

$$\dfrac{\Gamma, \psi[v/x], (RTC_{x,y}\,\varphi)(v,z) \vdash \Delta, \psi[z/x] \qquad \Gamma, \psi[z/x], \varphi[z/x, w/y] \vdash \Delta, \psi[w/x]}{\Gamma, \psi[v/x], (RTC_{x,y}\,\varphi)(v,z), \varphi[z/x, w/y] \vdash \Delta, \psi[w/x]}\ (\text{Cut})$$

$$\dfrac{\dfrac{}{\Gamma, \psi[v/x] \vdash \Delta, \psi[v/x]}\ (\text{Ax})}{\Gamma, \psi[v/x], v = w \vdash \Delta, \psi[w/x]}\ (=\text{L})$$

$$\dfrac{\Gamma, \psi[v/x], v = w \vdash \Delta, \psi[w/x] \qquad \begin{array}{c}\vdots\end{array}}{\Gamma, \psi[v/x], (RTC_{x,y}\,\varphi)(v,w) \vdash \Delta, \psi[w/x]}\ (\text{case-split})$$

$$\dfrac{\Gamma, \psi[v/x], (RTC_{x,y}\,\varphi)(v,w) \vdash \Delta, \psi[w/x]}{\Gamma, \psi[s/x], (RTC_{x,y}\,\varphi)(s,t) \vdash \Delta, \psi[t/x]}\ (\text{Subst})$$

$$\vdots$$

Implicit induction subsumes explicit induction

$$\dfrac{\Gamma, \psi[v/x], (RTC_{x,y}\,\varphi)(v,w) \vdash \Delta, \psi[w/x]}{\Gamma, \psi[v/x], (RTC_{x,y}\,\varphi)(v,z) \vdash \Delta, \psi[z/x]}\,(\text{Subst}) \qquad \dfrac{\begin{matrix}\vdots\\ \Gamma, \psi, \varphi \vdash \Delta, \psi[y/x]\end{matrix}}{\Gamma, \psi[z/x], \varphi[z/x, w/y] \vdash \Delta, \psi[w/x]}\,(\text{Subst})$$

$$\dfrac{\cdots}{\Gamma, \psi[v/x], (RTC_{x,y}\,\varphi)(v,z), \varphi[z/x, w/y] \vdash \Delta, \psi[w/x]}\,(\text{Cut})$$

$$\dfrac{\dfrac{\dfrac{}{\Gamma, \psi[v/x] \vdash \Delta, \psi[v/x]}\,(\text{Ax})}{\Gamma, \psi[v/x], v = w \vdash \Delta, \psi[w/x]}\,(=\text{L}) \qquad \begin{matrix}\Gamma, \psi[v/x], (RTC_{x,y}\,\varphi)(v,z), \varphi[z/x, w/y] \vdash \Delta, \psi[w/x]\\ \vdots\end{matrix}}{\Gamma, \psi[v/x], (RTC_{x,y}\,\varphi)(v,w) \vdash \Delta, \psi[w/x]}\,(\text{case-split})$$

$$\dfrac{\Gamma, \psi[v/x], (RTC_{x,y}\,\varphi)(v,w) \vdash \Delta, \psi[w/x]}{\begin{matrix}\Gamma, \psi[s/x], (RTC_{x,y}\,\varphi)(s,t) \vdash \Delta, \psi[t/x]\\ \vdots\end{matrix}}\,(\text{Subst})$$

$NCRTC^{\omega}_{G}$, the subsystem of non-overlapping cyclic proofs, is a Henkin-complete

Equivalence Under Arithmetic

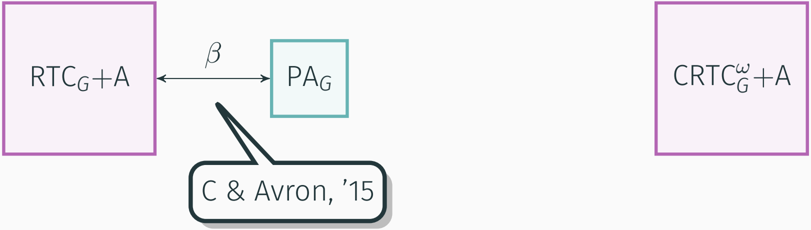
Obtain RTC_G+A and CRTC_G^ω+A by adding the following schemas:

1. $\mathsf{s}\,\mathbf{0} \vdash$
2. $\mathsf{s}\,x = \mathsf{s}\,y \vdash x = y$
3. $\vdash x + \mathbf{0} = x$
4. $\vdash x + \mathsf{s}\,y = \mathsf{s}\,(x + y)$
5. $\vdash (RTC_{v,w}\,\mathsf{s}\,v = w)(\mathbf{0}, x)$

Equivalence Under Arithmetic

Obtain RTC_G+A and $CRTC^{\omega}_G$+A by adding the following schemas:

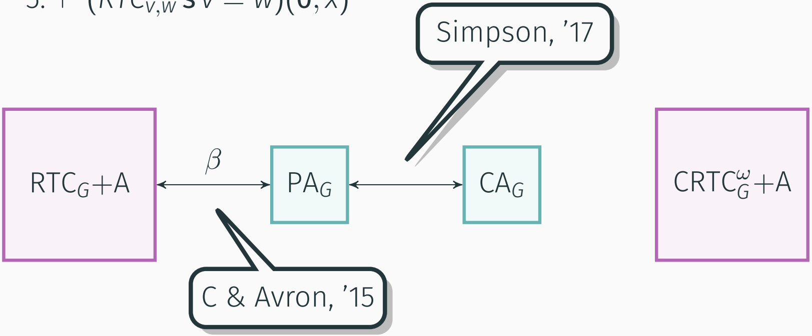
1. $\mathsf{s}\,\mathbf{0} \vdash$
2. $\mathsf{s}\,x = \mathsf{s}\,y \vdash x = y$
3. $\vdash x + \mathbf{0} = x$
4. $\vdash x + \mathsf{s}\,y = \mathsf{s}\,(x + y)$
5. $\vdash (RTC_{v,w}\,\mathsf{s}\,v = w)(\mathbf{0}, x)$



Equivalence Under Arithmetic

Obtain RTC_G+A and $CRTC_G^\omega$+A by adding the following schemas:

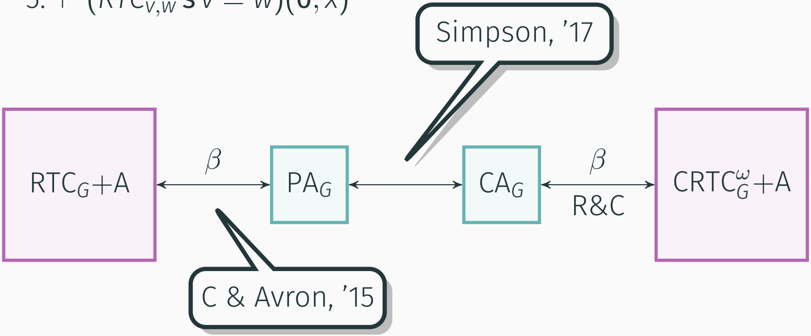
1. $\mathbf{s}\,\mathbf{0} \vdash$
2. $\mathbf{s}\,x = \mathbf{s}\,y \vdash x = y$
3. $\vdash x + \mathbf{0} = x$
4. $\vdash x + \mathbf{s}\,y = \mathbf{s}\,(x + y)$
5. $\vdash (RTC_{v,w}\,\mathbf{s}\,v = w)(\mathbf{0}, x)$



Equivalence Under Arithmetic

Obtain RTC_G+A and $CRTC^{\omega}_G$+A by adding the following schemas:

1. $\mathsf{s}\,\mathbf{0} \vdash$
2. $\mathsf{s}\,x = \mathsf{s}\,y \vdash x = y$
3. $\vdash x + \mathbf{0} = x$
4. $\vdash x + \mathsf{s}\,y = \mathsf{s}\,(x + y)$
5. $\vdash (RTC_{v,w}\,\mathsf{s}\,v = w)(\mathbf{0}, x)$



Equivalence: The General Case

For $\mathsf{FOL}_{\mathsf{ID}}$, implicit (cyclic) induction generally stronger than explicit induction [Berardi & Tatsuta, '17]

Equivalence: The General Case

For $\mathsf{FOL}_{\mathsf{ID}}$, implicit (cyclic) induction generally stronger than explicit induction [Berardi & Tatsuta, '17]

- For signature $\{0, s\} + \{N\}$:
  - 0,s-axioms $\vdash_{\mathsf{CLKID}^{\omega}}$ "2-hydra"

For $\mathsf{FOL}_{\mathsf{ID}}$, implicit (cyclic) induction generally stronger than explicit induction [Berardi & Tatsuta, '17]

- For signature $\{0, s\} + \{N\}$:
  - 0,s-axioms $\vdash_{\mathrm{CLKID}^{\omega}}$ "2-hydra"
  - 0,s-axioms $\nvdash_{\mathrm{LKID}}$ "2-hydra"
    (Henkin counter-model construction)

Equivalence: The General Case

For $\mathsf{FOL}_{\mathsf{ID}}$, implicit (cyclic) induction generally stronger than explicit induction [Berardi & Tatsuta, '17]

- For signature $\{0, s\} + \{N\}$:
  - 0,s-axioms $\vdash_{\mathsf{CLKID}^{\omega}}$ "2-hydra"
  - 0,s-axioms $\nvdash_{\mathsf{LKID}}$ "2-hydra"
    (Henkin counter-model construction)
- However, for signature $\{0, s\} + \{N, \leq\}$
  - 0,s-axioms $\vdash_{\mathsf{LKID}}$ 2-hydra

Equivalence: The General Case

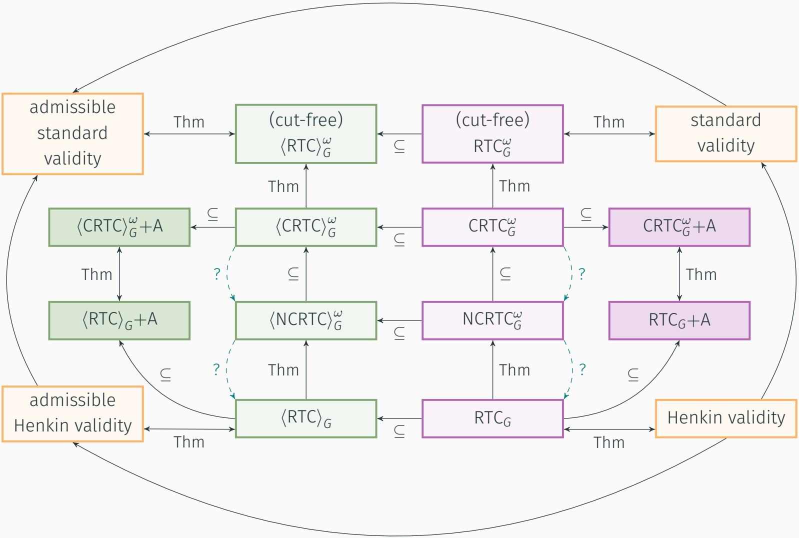
For $\mathsf{FOL}_{\mathsf{ID}}$, implicit (cyclic) induction generally stronger than explicit induction [Berardi & Tatsuta, '17]

- For signature $\{0, s\} + \{N\}$:
  - 0,s-axioms $\vdash_{\mathsf{CLKID}^\omega}$ "2-hydra"
  - 0,s-axioms $\nvdash_{\mathsf{LKID}}$ "2-hydra"
    (Henkin counter-model construction)
- However, for signature $\{0, s\} + \{N, \leq\}$
  - 0,s-axioms $\vdash_{\mathsf{LKID}}$ 2-hydra

So this does not serve to show RTC_G and CRTC^ω_G inequivalent

- **TC** has all inductive definitions available

Summary of Results



Future Work

- Resolving the open question of the (in)equivalence of RTC_G, NCRTC_G^ω and CRTC_G^ω.
- Implementing CRTC_G^ω 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
- A uniform framework for coinductive reasoning?

Recall transitive closure as a fixed point:

$$R^{+} = \mu X.\Psi_R(X) \qquad\qquad \Psi_R(S) = R \cup (R \circ S)$$

The greatest fixed point gives the transitive co-closure

- Pairs (s, t) in $\nu X.\Psi_R(X)$ are those connected by a possibly infinite number of R-steps
- We can write $(RTC^{\mathrm{op}}_{x,y}\,\varphi)(s, t)$ to denote that (s, t) is in the reflexive, transitive co-closure of φ
- E.g. The following formula defines possibly infinite lists

$$(RTC^{\mathrm{op}}_{x,y}\,\exists z\,.\,x = \mathsf{cons}(z, y))(v, [\,])$$

We have the following standard semantics

$$M, v \models (RTC^{\mathrm{op}}_{x,y}\,\varphi)(s, t) \Leftrightarrow$$
$$\exists(\vec{a}_i)_{i\geq 0}\,.\,\forall i \geq 0\,.\,a_i = v(t) \vee M, v[x := a_i, y := a_{i+1}] \models \varphi$$

We have the following Henkin-semantics

$$H, v \models_{\mathcal{H}} (RTC^{\mathrm{op}}_{x,y}\,\varphi)(s, t) \Leftrightarrow$$
$$\text{there exists } A \in \mathcal{D} \text{ such that } v(s) \in A \text{ and}$$
$$\forall a \in A\,.\,\text{either } a = v(t) \text{ or } \exists b \in A\,.\,H, v[x := a, y := b] \models_{\mathcal{H}} \varphi$$