Infinitary and Cyclic Proof Systems for Transitive Closure Logic

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

[1] Dept. of Computer Science, Cornell University, NY, USA, `lironcohen@cornell.edu`
[2] School of Computing, University of Kent, Canterbury, UK, `r.n.s.rowe@kent.ac.uk`

Abstract. Transitive closure logic is a known extension of first-order logic obtained by introducing a transitive closure operator. While other extensions of first-order logic with inductive definitions are a priori parametrized by a set of inductive definitions, the addition of the transitive closure operator uniformly captures all finitary inductive definitions. In this paper we present an *infinitary* proof system for transitive closure logic which is an *infinite descent*-style counterpart to the existing (explicit induction) proof system for the logic. We show that, as for similar systems for first-order logic with inductive definitions, our infinitary system is complete for the standard semantics and subsumes the explicit system. Moreover, the uniformity of the transitive closure operator allows semantically meaningful complete restrictions to be defined using simple syntactic criteria. Consequently, the restriction to regular infinitary (i.e. *cyclic*) proofs provides the basis for an effective system for automating inductive reasoning.

1 Introduction

A core technique in mathematical reasoning is that of *induction*. This is especially true in computer science, where it plays a central role in reasoning about recursive data and computations. Formal systems for mathematical reasoning usually capture the notion of inductive reasoning via one or more inference rules that express the general induction schemes, or principles, that hold for the elements being reasoned over.

Increasingly, we are concerned with not only being able to formalise as much mathematical reasoning as possible, but also with doing so in an effective way. In other words, we seek to be able to automate such reasoning. *Transitive closure* (TC) logic has been identified as a potential candidate for a minimal, 'most general' system for inductive reasoning, which is also very suitable for automation [1,10,11]. TC adds to first-order logic a single operator for forming binary relations: specifically, the transitive closures of arbitrary formulas (more precisely, the transitive closure of the binary relation induced by a formula with respect to two distinct variables). In this work, for simplicity, we use a reflexive form of the operator; however the two forms are equivalent in the presence of equality. This modest addition affords enormous expressive power: namely it provides a uniform way of capturing inductive principles. If an induction scheme

is expressed by a formula φ, then the elements of the inductive collection it defines are those 'reachable' from the base elements x via the iteration of the induction scheme. That is, those y's for which (x, y) is in the transitive closure of φ. Thus, bespoke induction principles do not need to be added to, or embedded within, the logic; instead, all induction schemes are available within a single, unified language. In this respect, the transitive closure operator resembles the W-type [21], which also provides a single type constructor from which one can uniformly define a variety of inductive types.

TC logic is intermediate between first- and second-order logic. Furthermore, since the TC operator is a particular instance of a least fixed point operator, TC logic is also subsumed by fixed-point logics such as the μ-calculus [18]. However, despite its minimality TC logic retains enough expressivity to capture inductive reasoning, as well as to subsume arithmetics (see Section 4.2.1). Moreover, from a proof theoretical perspective the conciseness of the logic makes it of particular interest. The use of only one constructor of course comes with a price: namely, formalizations (mostly of non-linear induction schemes) may be somewhat complex. However, they generally do not require as complex an encoding as in arithmetics, since the TC operator can be applied on any formula and thus (depending on the underlying signature) more naturally encode induction on sets more complex than the natural numbers.

Since its expressiveness entails that TC logic subsumes arithmetics, by Gödel's result, any effective proof system for it must necessarily be incomplete for the standard semantics. Notwithstanding, a natural, effective proof system which is sound for TC logic was shown to be complete with respect to a generalized form of Henkin semantics [9]. In this paper, following similar developments in other formalizations for fixed point logics and inductive reasoning (see e.g. [4,5,6,23,25]), we present an *infinitary* proof theory for TC logic which, as far as we know, is the first system that is (cut-free) complete with respect to the standard semantics. More specifically, our system employs infinite-height, rather than infinite-width proofs (see Section 3.2). The soundness of such infinitary proof theories is underpinned by the principle of *infinite descent*: proofs are permitted to be infinite, non-well-founded trees, but subject to the restriction that every infinite path in the proof admits some infinite descent. The descent is witnessed by tracing terms or formulas for which we can give a correspondence with elements of a well-founded set. In particular, we can trace terms that denote elements of an inductively defined (well-founded) set. For this reason, such theories are considered systems of implicit induction, as opposed to those which employ explicit rules for applying induction principles. While a full infinitary proof theory is clearly not effective, in the aforementioned sense, such a system can be obtained by restricting consideration to only the *regular* infinite proofs. These are precisely those proofs that can be finitely represented as (possibly cyclic) graphs.

These infinitary proof theories generally subsume systems of explicit induction in expressive power, but also offer a number of advantages. Most notably, they can ameliorate the primary challenge for inductive reasoning: finding an induction

invariant. In explicit induction systems, this must be provided *a priori*, and is often much stronger than the goal one is ultimately interested in proving. However, in implicit systems the inductive arguments and hypotheses may be encoded in the cycles of a proof, so cyclic proof systems seem better for automation. The cyclic approach has also been used to provide an optimal cut-free complete proof system for Kleene algebra [15], providing further evidence of its utility for automation.

In the setting of TC logic, we observe some further benefits over more traditional formal systems of inductive definitions and their infinitary proof theories (cf. LKID [6,20]). TC (with a pairing function) has all first-order definable finitary inductive definitions immediately 'available' within the language of the logic: as with inductive hypotheses, one does not need to 'know' in advance which induction schemes will be required. Moreover, the use of a single transitive closure operator provides a uniform treatment of all induction schemes. That is, instead of having a proof system parameterized by a set of inductive predicates and rules for them (as is the case in LKID), TC offers a single proof system with a single rule scheme for induction. This has immediate advantages for developing the metatheory: the proofs of completeness for standard semantics and adequacy (i.e. subsumption of explicit induction) for the infinitary system presented in this paper are simpler and more straightforward. Moreover, it permits a cyclic subsystem, which also subsumes explicit induction, to be defined via a simple syntactic criterion that we call *normality*. The smaller search space of possible proofs further enhances the potential for automation. TC logic seems more expressive in other ways, too. For instance, the transitive closure operator may be applied to arbitrarily complex formulas, not only to collections of atomic formulas (cf. Horn clauses), as in e.g. [4,6].

We show that the explicit and cyclic TC systems are equivalent under arithmetic, as is the case for LKID [3,24]. However, there are cases in which the cyclic system for LKID is strictly more expressive than the explicit induction system [2]. To obtain a similar result for TC, the fact that all induction schemes are available poses a serious challenge. For one, the counter-example used in [2] does not serve to show this result holds for TC. If this strong inequivalence indeed holds also for TC, it must be witnessed by a more subtle and complex counter-example. Conversely, it may be that the explicit and cyclic systems do coincide for TC. In either case, this points towards fundamental aspects that require further investigation.

The rest of the paper is organised as follows. In Section 2 we reprise the definition of transitive closure logic and both its standard and Henkin-style semantics. Section 3 presents the existing explicit induction proof system for TC logic, and also our new infinitary proof system. We prove the latter sound and complete for the standard semantics, and also derive cut-admissibility. In Section 4 we compare the expressive power of the infinitary system (and its cyclic subsystem) with the explicit system. Section 5 concludes and examines the remaining open questions for our system as well as future work.

3

This technical report comprises an extended version, with proofs, of the results presented in [12]. We would like to thank an anonymous reviewer for bringing to our attention a technical problem with the proof of completeness for the infinitary system in a previous version of this work.

2 Transitive Closure Logic and its Semantics

In this section we review the language of transitive closure logic, and two possible semantics for it: a standard one, and a Henkin-style one. For simplicity of presentation we assume (as is standard practice) a designated equality symbol in the language. We denote by $v[x_1 := a_n, \ldots, x_n := a_n]$ the variant of the assignment v which assigns a_i to x_i for each i, and by $\varphi\left\{\frac{t_1}{x_1}, \ldots, \frac{t_n}{x_n}\right\}$ the result of simultaneously substituting each t_i for the free occurrences of x_i in φ.

Definition 1 (The language $\mathcal{L}_{\mathsf{RTC}}$). *Let σ be a first-order signature with equality, whose terms are ranged over by s and t and predicates by P, and let x, y, z, etc. range over a countable set of variables. The language $\mathcal{L}_{\mathsf{RTC}}$ consists of the formulas defined by the grammar:*

$$\varphi, \psi ::= s = t \mid P(t_1, \ldots, t_n) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \to \varphi \mid \forall x.\varphi \mid \exists x.\varphi \mid$$
$$(RTC_{x,y}\,\varphi)(s, t)$$

As usual, $\forall x$ and $\exists x$ bind free occurrences of the variable x and we identify formulas up to renaming of bound variables, so that capturing of free variables during substitution does not occur. Note that in the formula $(RTC_{x,y}\,\varphi)(s, t)$ free occurrences of x and y in φ are also bound (but not those in s and t).

Definition 2 (Standard Semantics). *Let $M = \langle D, I \rangle$ be a first-order structure (i.e. D is a non-empty domain and I an interpretation function), and v an assignment in M which we extend to terms in the obvious way. The satisfaction relation $\models$ between model-valuation pairs $\langle M, v \rangle$ and formulas is defined inductively on the structure of formulas by:*

- $M, v \models s = t$ *if* $v(s) = v(t)$;
- $M, v \models P(t_1, \ldots, t_n)$ *if* $(v(t_1), \ldots, v(t_n)) \in I(P)$;
- $M, v \models \neg\varphi$ *if* $M, v \not\models \varphi$;
- $M, v \models \varphi_1 \wedge \varphi_2$ *if both* $M, v \models \varphi_1$ *and* $M, v \models \varphi_2$;
- $M, v \models \varphi_1 \vee \varphi_2$ *if either* $M, v \models \varphi_1$ *or* $M, v \models \varphi_2$;
- $M, v \models \varphi_1 \to \varphi_2$ *if* $M, v \models \varphi_1$ *implies* $M, v \models \varphi_2$;
- $M, v \models \exists x.\varphi$ *if* $M, v[x := a] \models \varphi$ *for some* $a \in D$;
- $M, v \models \forall x.\varphi$ *if* $M, v[x := a] \models \varphi$ *for all* $a \in D$;
- $M, v \models (RTC_{x,y}\,\varphi)(s, t)$ *if* $v(s) = v(t)$, *or there exist* $a_0, \ldots, a_n \in D$ $(n > 0)$ *s.t.* $v(s) = a_0$, $v(t) = a_n$, *and* $M, v[x := a_i, y := a_{i+1}] \models \varphi$ *for* $0 \leq i < n$.

We say that a formula φ is valid with respect to the standard semantics when $M, v \models \varphi$ holds for all models M and valuations v.

4

We next recall the concepts of frames and Henkin structures (see, e.g., [17]). A frame is a first-order structure together with some subset of the powerset of its domain (called its set of admissible subsets).

Definition 3 (Frames). *A frame M is a triple $\langle D, I, \mathcal{D} \rangle$, where $\langle D, I \rangle$ is a first-order structure, and $\mathcal{D} \subseteq \wp(D)$.*

Note that if $\mathcal{D} = \wp(D)$, the frame is identified with a standard first-order structure.

Definition 4 (Frame Semantics). *$\mathcal{L}_{\mathsf{RTC}}$ formulas are interpreted in frames as in Definition 2 above, except for:*

- *$M, v \models (RTC_{x,y}\, \varphi)(s,t)$ if for every $A \in \mathcal{D}$, if $v(s) \in A$ and for every $a, b \in D$: $a \in A$ and $M, v[x := a, y := b] \models \varphi$ implies $b \in A$, then $v(t) \in A$.*

We now consider Henkin structures, which are frames whose set of admissible subsets is closed under parametric definability.

Definition 5 (Henkin structures). *A Henkin structure $M = \langle D, I, \mathcal{D} \rangle$ is a frame such that $\{a \in D \mid M, v[x := a] \models \varphi\} \in \mathcal{D}$ for every φ, and v in M.*

We refer to the semantics induced by quantifying over the (larger) class of Henkin structures as the Henkin semantics.

It is worth noting that the inclusion of equality in the basic language is merely for notational convenience. This is because the RTC operator allows us, under both the standard and Henkin semantics, to actually *define* equality $s = t$ on terms as $(RTC_{x,y} \perp)(s,t)$.

3 Proof Systems for $\mathcal{L}_{\mathsf{RTC}}$

In this section, we define two proof systems for $\mathcal{L}_{\mathsf{RTC}}$. The first is a finitary proof system with an explicit induction rule for RTC formulas. The second is an infinitary proof system, in which RTC formulas are simply unfolded, and inductive arguments are represented via infinite descent-style constructions. We show the soundness and completeness of these proof systems, and also compare their provability relations.

Our systems for $\mathcal{L}_{\mathsf{RTC}}$ are extensions of $\mathcal{LK}_=$, the sequent calculus for classical first-order logic with equality [16,26] whose proof rules we show in Fig. 1.[3] Sequents are expressions of the form $\Gamma \Rightarrow \Delta$, for finite sets of formulas Γ and Δ. We write Γ, Δ and Γ, φ as a shorthand for $\Gamma \cup \Delta$ and $\Gamma \cup \{\varphi\}$ respectively, and $\mathsf{fv}(\Gamma)$ for the set of free variables of the formulas in the set Γ. A sequent $\Gamma \Rightarrow \Delta$ is valid if and only if the formula $\bigwedge_{\varphi \in \Gamma} \varphi \to \bigvee_{\psi \in \Delta} \psi$ is.

---

[3] Here we take $\mathcal{LK}_=$ to include the substitution rule, which was not a part of the original systems.

$$(\text{Axiom}): \dfrac{}{\varphi \Rightarrow \varphi} \qquad (\text{WL}): \dfrac{\Gamma \Rightarrow \Delta}{\Gamma, \varphi \Rightarrow \Delta} \qquad (\text{WR}): \dfrac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \varphi}$$

$$(\vee\text{L}): \dfrac{\Gamma \Rightarrow \varphi, \Delta \quad \Gamma, \psi \Rightarrow \Delta}{\Gamma, \varphi \vee \psi \Rightarrow \Delta} \quad (\wedge\text{L}): \dfrac{\Gamma, \varphi, \psi \Rightarrow \Delta}{\Gamma, \varphi \wedge \psi \Rightarrow \Delta} \quad (\to\text{L}): \dfrac{\Gamma \Rightarrow \varphi, \Delta \quad \Gamma, \psi \Rightarrow \Delta}{\Gamma, \varphi \to \psi \Rightarrow \Delta}$$

$$(\vee\text{R}): \dfrac{\Gamma \Rightarrow \varphi, \psi, \Delta}{\Gamma \Rightarrow \varphi \vee \psi, \Delta} \quad (\wedge\text{R}): \dfrac{\Gamma \Rightarrow \varphi, \Delta \quad \Gamma \Rightarrow \psi, \Delta}{\Gamma \Rightarrow \varphi \wedge \psi, \Delta} \quad (\to\text{R}): \dfrac{\Gamma, \varphi \Rightarrow \psi, \Delta}{\Gamma \Rightarrow \varphi \to \psi, \Delta}$$

$$(\exists\text{L}): \dfrac{\Gamma, \varphi \Rightarrow \Delta}{\Gamma, \exists x.\varphi \Rightarrow \Delta} \, x \notin \mathsf{fv}(\Gamma, \Delta) \quad (\forall\text{L}): \dfrac{\Gamma, \varphi\left\{\frac{t}{x}\right\} \Rightarrow \Delta}{\Gamma, \forall x.\varphi \Rightarrow \Delta} \quad (\neg\text{L}): \dfrac{\Gamma \Rightarrow \varphi, \Delta}{\Gamma, \neg\varphi \Rightarrow \Delta}$$

$$(\exists\text{R}): \dfrac{\Gamma \Rightarrow \varphi\left\{\frac{t}{x}\right\}, \Delta}{\Gamma \Rightarrow \exists x.\varphi, \Delta} \quad (\forall\text{R}): \dfrac{\Gamma \Rightarrow \varphi, \Delta}{\Gamma \Rightarrow \forall x.\varphi, \Delta} \, x \notin \mathsf{fv}(\Gamma, \Delta) \quad (\neg\text{R}): \dfrac{\Gamma, \varphi \Rightarrow \Delta}{\Gamma \Rightarrow \neg\varphi, \Delta}$$

$$(=\text{L}_1): \dfrac{\Gamma \Rightarrow \varphi\left\{\frac{s}{x}\right\}, \Delta}{\Gamma, s = t \Rightarrow \varphi\left\{\frac{t}{x}\right\}, \Delta} \quad (=\text{L}_2): \dfrac{\Gamma \Rightarrow \varphi\left\{\frac{t}{x}\right\}, \Delta}{\Gamma, s = t \Rightarrow \varphi\left\{\frac{s}{x}\right\}, \Delta} \quad (=\text{R}): \dfrac{}{\Rightarrow t = t}$$

$$(\text{Cut}): \dfrac{\Gamma \Rightarrow \varphi, \Delta \quad \Sigma, \varphi \Rightarrow \Pi}{\Gamma, \Sigma \Rightarrow \Delta, \Pi} \quad (\text{Subst}): \dfrac{\Gamma \Rightarrow \Delta}{\Gamma\left\{\frac{t_1}{x_1}, \ldots, \frac{t_n}{x_n}\right\} \Rightarrow \Delta\left\{\frac{t_1}{x_1}, \ldots, \frac{t_n}{x_n}\right\}}$$

Fig. 1: Proof rules for the sequent calculus $\mathcal{LK}_=$ with substitution.

3.1 The Finitary Proof System

We briefly summarise the finitary proof system for $\mathcal{L}_{\mathsf{RTC}}$. For more details see [10,11]. We write $\varphi(x_1, \ldots, x_n)$ to emphasise that the formula φ may contain $x_1, \ldots, x_n$ as free variables.

Definition 6. *The proof system* RTC_G *for* $\mathcal{L}_{\mathsf{RTC}}$ *is defined by adding to* $\mathcal{LK}_=$ *the following inference rules:*

$$\dfrac{}{\Gamma \Rightarrow \Delta, (RTC_{x,y}\, \varphi)(s, s)} \tag{1}$$

$$\dfrac{\Gamma \Rightarrow \Delta, (RTC_{x,y}\, \varphi)(s, r) \quad \Gamma \Rightarrow \Delta, \varphi\left\{\frac{r}{x}, \frac{t}{y}\right\}}{\Gamma \Rightarrow \Delta, (RTC_{x,y}\, \varphi)(s, t)} \tag{2}$$

$$\dfrac{\Gamma, \psi(x), \varphi(x, y) \Rightarrow \Delta, \psi\left\{\frac{y}{x}\right\}}{\Gamma, \psi\left\{\frac{s}{x}\right\}, (RTC_{x,y}\, \varphi)(s, t) \Rightarrow \Delta, \psi\left\{\frac{t}{x}\right\}} \tag{3}$$

where, for Rule (3), $x \notin \mathsf{fv}(\Gamma, \Delta)$ and $y \notin \mathsf{fv}(\Gamma, \Delta, \psi)$

Rule (3) is a generalized induction principle. It states that if an extension of formula ψ is closed under the relation induced by φ, then it is also closed under

the reflexive transitive closure of that relation. In the case of arithmetic this rule captures the induction rule of Peano's Arithmetics PA [11].

3.2 Infinitary Proof Systems

Definition 7. *The infinitary proof system* RTC_G^ω *for* $\mathcal{L}_{\mathsf{RTC}}$ *is defined like* RTC_G, *but replacing Rule (3) by:*

$$\frac{\Gamma, s = t \Rightarrow \Delta \quad \Gamma, (RTC_{x,y}\,\varphi)(s,z), \varphi\left\{\frac{z}{x}, \frac{t}{y}\right\} \Rightarrow \Delta}{\Gamma, (RTC_{x,y}\,\varphi)(s,t) \Rightarrow \Delta} \tag{4}$$

where z is fresh, i.e. does not occur free in Γ, Δ, or $(RTC_{x,y}\,\varphi)(s,t)$. The formula $(RTC_{x,y}\,\varphi)(s,z)$ in the right-hand premise is called the immediate ancestor *(cf. [7, §1.2.3]) of the principal formula, $(RTC_{x,y}\,\varphi)(s,t)$, in the conclusion.*

There is an asymmetry between Rule (2), in which the intermediary is an arbitrary term r, and Rule (4), where we use a variable z. This is necessary to obtain the soundness of the cyclic proof system. It is used to show that when there is a counter-model for the conclusion of a rule, then there is also a counter-model for one of its premises that is, in a sense that we make precise below, 'smaller'. In the case that $s \neq t$, using a fresh z allows us to pick from *all* possible counter-models of the conclusion, from which we may then construct the required counter-model for the right-hand premise. If we allowed an arbitrary term r instead, this might restrict the counter-models we can choose from, only leaving ones 'larger' than the one we had for the conclusion. See Lemma 1 below for more details.

Proofs in this system are possibly infinite derivation trees. However, not all infinite derivations are proofs: only those that admit an infinite descent argument. Thus we use the terminology 'pre-proof' for derivations.

Definition 8 (Pre-proofs). *An* RTC_G^ω *pre-proof is a possibly infinite (i.e. non-well-founded) derivation tree formed using the inference rules. A* path *in a pre-proof is a possibly infinite sequence of sequents $s_0, s_1, \ldots (, s_n)$ such that s_0 is the root sequent of the proof, and s_{i+1} is a premise of s_i for each $i < n$.*

The following definitions tell us how to track *RTC* formulas through a pre-proof, and allow us to formalize inductive arguments via infinite descent.

Definition 9 (Trace Pairs). *Let τ and τ' be RTC formulas occurring in the left-hand side of the conclusion s and a premise s', respectively, of (an instance of) an inference rule. (τ, τ') is said to be a* trace pair *for (s, s') if the rule is:*

- *the (Subst) rule, and $\tau = \tau'\theta$ where θ is the substitution associated with the rule instance;*
- *Rule (4), and either:*
  - *a) τ is the principal formula of the rule instance and τ' is the immediate ancestor of τ, in which case we say that the trace pair is* progressing;

7

b) otherwise, $\tau = \tau'$.
− *any other rule, and $\tau = \tau'$.*

Definition 10 (Traces). *A* trace *is a (possibly infinite) sequence of RTC formulas. We say that a trace $\tau_1, \tau_2, \ldots (, \tau_n)$ follows a path $s_1, s_2, \ldots (, s_m)$ in a pre-proof $\mathcal{P}$ if, for some $k \geq 0$, each consecutive pair of formulas (τ_i, τ_{i+1}) is a trace pair for (s_{i+k}, s_{i+k+1}). If (τ_i, τ_{i+1}) is a progressing pair then we say that the trace* progresses *at i, and we say that the trace is* infinitely progressing *if it progresses at infinitely many points.*

Proofs, then, are pre-proofs which satisfy a global trace condition.

Definition 11 (Infinite Proofs). *A RTC_G^ω proof is a pre-proof in which every infinite path is followed by some infinitely progressing trace.*

Clearly, we cannot reason effectively about such infinite proofs in general. In order to do so we need to restrict our attention to those proof trees which are finitely representable. These are the *regular* infinite proof trees, which contain only finitely many *distinct* subtrees. They can be specified as systems of recursive equations or, alternatively, as cyclic *graphs* [14]. Note that a given regular infinite proof may have many different graph representations. One possible way of formalizing such proof graphs is as standard proof trees containing open nodes (called buds), to each of which is assigned a syntactically equal internal node of the proof (called a companion). Due to space limitation, we elide a formal definition of cyclic proof graphs (see, e.g., Sect. 7 in [6]) and rely on the reader's basic intuitions.

Definition 12 (Cyclic Proofs). *The cyclic proof system CRTC_G^ω for $\mathcal{L}_{\mathsf{RTC}}$ is the subsystem of RTC_G^ω comprising of all and only the finite and* regular *infinite proofs (i.e. those proofs that can be represented as finite, possibly cyclic, graphs).*

Note that it is decidable whether a cyclic pre-proof satisfies the global trace condition, using a construction involving an inclusion between Büchi automata (see, e.g., [4,24]). However since this requires complementing Büchi automata (a PSPACE procedure), our system cannot be considered a proof system in the Cook-Reckhow sense [13]. Notwithstanding, checking the trace condition for cyclic proofs found in practice is not prohibitive [22,27].

3.3 Soundness and Completeness

The rich expressiveness of TC logic entails that the effective system RTC_G which is sound w.r.t. the standard semantics, cannot be complete (much like the case for LKID). It is however both sound and complete w.r.t. Henkin semantics.

Theorem 1 (Soundness and Completeness of RTC_G [9]). *RTC_G is sound for standard semantics, and also sound and complete for Henkin semantics.*

Note that the system RTC_G as presented here does not admit cut elimination. The culprit is the induction rule (3), which does not permute with cut. We may obtain admissibility of cut by using the following alternative formulation of the induction rule:

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

where $x \notin \mathsf{fv}(\Gamma, \Delta)$, $y \notin \mathsf{fv}(\Gamma, \Delta, \psi)$. Like the induction rule for LKID, this formulation incorporates a cut with the induction formula ψ. For the system with this rule, a simple adaptation of the completeness proof in [9], in the spirit of the corresponding proof for LKID in [6], suffices to obtain cut-free completeness. However, the tradeoff is that the resulting cut-free system no longer has the sub-formula property. In contrast, cut-free proofs in RTC_G do satisfy the sub-formula property, for a generalized notion of a subformula that incorporates substitution instances (as in $\mathcal{LK}_=$).

We remark that the soundness proof of LKID is rather complex since it must handle different types of mutual dependencies between the inductive predicates. For RTC_G the proof is much simpler due to the uniformity of the rules for the RTC operator.

The infinitary system RTC_G^ω, in contrast to the finitary system RTC_G, is both sound and complete w.r.t. the standard semantics. To prove soundness, we make use of the following notion of *measure* for RTC formulas.

Definition 13 (Degree of RTC Formulas). *For $\phi \equiv (RTC_{x,y}\,\varphi)(s,t)$, we define $\delta_\phi(M,v) = 0$ if $v(s) = v(t)$, and $\delta_\phi(M,v) = n$ if $v(s) \neq v(t)$ and $a_0, \ldots, a_n$ is a minimal-length sequence of elements in the semantic domain D such that $v(s) = a_0$, $v(t) = a_n$, and $M, v[x := a_i, y := a_{i+1}] \models \varphi$ for $0 \leq i < n$. We call $\delta_\phi(M,v)$ the* degree *of ϕ with respect to the model M and valuation v.*

Soundness then follows from the following fundamental lemma.

Lemma 1 (Descending Counter-models). *If there exists a standard model M and valuation v that invalidates the conclusion s of (an instance of) an inference rule, then 1) there exists a standard model M' and valuation v' that invalidates some premise s' of the rule; and 2) if (τ, τ') is a trace pair for (s, s') then $\delta_{\tau'}(M',v') \leq \delta_\tau(M,v)$. Moreover, if (τ, τ') is a progressing trace pair then $\delta_{\tau'}(M',v') < \delta_\tau(M,v)$.*

Proof. The cases for the standard $\mathcal{LK}_=$ and substitution rules are straightforward adaptations of those found in e.g. [6].

– The case for Rule (1) follows trivially since it follows immediately from Definition 2 that $M, v \models (RTC_{x,y}\,\varphi)(s,s)$ for all M and v.

– For Rule (2), since $M, v \not\models (RTC_{x,y}\,\varphi)(s,t)$ it follows that either $M, v \not\models (RTC_{x,y}\,\varphi)(s,r)$ or $M, v \not\models \varphi\left\{\frac{r}{x}, \frac{t}{y}\right\}$. To see this, suppose for contradiction that both $M, v \models (RTC_{x,y}\,\varphi)(s,r)$ or $M, v \models \varphi\left\{\frac{r}{x}, \frac{t}{y}\right\}$; but then it would

follow by Definition 2 that $M, v \models (RTC_{x,y}\, \varphi)(s,t)$. We thus take $M' = M$ and $v' = v$, and either the left- or right-hand premise according to whether $M, v \not\models (RTC_{x,y}\, \varphi)(s,r)$ or $M, v \not\models \varphi\left\{\frac{r}{x}, \frac{t}{y}\right\}$.

 – For Rule (4), since $M, v \models (RTC_{x,y}\, \varphi)(s,t)$ there are two cases to consider:

(i) If $v(s) = v(t)$ then we take the left-hand premise with model $M' = M$ and valuation $v' = v$, and so the degree of any RTC formula in Γ with respect to M' and v' remains the same.

(ii) If on the other hand there are $a_0, \ldots, a_n \in D$ $(n > 0)$ such that $v(s) = a_0$ and $v(t) = a_n$ with $M, v[x := a_i, y := a_{i+1}] \models \varphi$ for $0 \leq i < n$, we then take the right-hand premise, the model $M' = M$ and valuation $v' = v[z := a_{n-1}]$. Note that, without loss of generality, we may assume a sequence $a_0, \ldots, a_n$ of minimal length, and thus surmise $\delta_{(RTC_{x,y}\, \varphi)(s,t)}(M, v) = n$. Since z is fresh, it follows that $M', v' \models \varphi\left\{\frac{z}{x}, \frac{t}{y}\right\}$ and $M', v'[x := a_i, y := a_{i+1}] \models \varphi$ for $0 \leq i < n - 1$. If $n = 1$ then $v'(s) = v'(z) = a_0$ and so $M, v' \models (RTC_{x,y}\, \varphi)(s,z)$; otherwise this is witnessed by the sequence $a_0, \ldots, a_{n-1}$. Thus we also have that $\delta_{(RTC_{x,y}\, \varphi)(s,z)}(M', v') = n - 1$. To conclude, note it also follows from z fresh that $M', v' \models \psi$ for all $\psi \in \Gamma$ and $M', v' \not\models \phi$ for all $\phi \in \Delta$; and furthermore that the degree of any RTC formula in Γ remains unchanged with respect to M' and v'. $\qquad\square$

As is standard for infinite descent inference systems [4,5,6,15,22,27], the above result entails the local soundness of the inference rules (in our case, for standard first-order models). The presence of infinitely progressing traces for each infinite path in a RTC_G^ω proof ensures soundness via a standard infinite descent-style construction.

Theorem 2 (Soundness of RTC_G^ω). *If there is a RTC_G^ω proof of $\Gamma \Rightarrow \Delta$, then $\Gamma \Rightarrow \Delta$ is valid (w.r.t. the standard semantics)*

Proof. Suppose, for contradiction, that $\Gamma \Rightarrow \Delta$ is not valid. Then by Lemma 1 there exists an infinite path $\{s_i\}_{i>0}$ in the proof and an infinite sequence of model-valuation pairs $\{\langle M_i, v_i \rangle\}_{i>0}$ such that $\langle M_i, v_i \rangle$ invalidates s_i for each $i > 0$. Since the proof is a valid RTC_G^ω proof, this infinite path is followed by an infinitely progressing trace $\{\tau_i\}_{i>0}$ for which we can take the degree of each formula with respect to its corresponding counter-model to obtain an infinite sequence of natural numbers $\{\delta_{\tau_i}(M_{k+i}, v_{k+i})\}_{i>0}$ (for some $k \geq 0$). By Lemma 1 this sequence is decreasing and, moreover, since the trace is infinitely progressing the sequence strictly decreases infinitely often. From the fact that the natural numbers are a well-founded set we derive a contradiction, and thus conclude that $\Gamma \Rightarrow \Delta$ is indeed valid. $\qquad\square$

The soundness of the cyclic system is an immediate corollary, since each CRTC_G^ω proof is also a RTC_G^ω proof.

Corollary 1 (Soundness of CRTC_G^ω). *If there is a CRTC_G^ω proof of $\Gamma \Rightarrow \Delta$, then $\Gamma \Rightarrow \Delta$ is valid (w.r.t. the standard semantics)*

Following a standard technique (as used in e.g. [6]), we can show cut-free completeness of RTC_G^ω with respect to the standard semantics.

Definition 14 (Schedule). *A* schedule element *E is defined as any of the following:*

- *a formula of the form $\neg\varphi, \varphi \wedge \psi, \varphi \vee \psi, \varphi \to \psi$;*
- *a pair of the form $\langle \forall x\, \varphi, t \rangle$ or $\langle \exists x\, \varphi, t \rangle$ where $\forall x\, \varphi$ and $\exists x\, \varphi$ are formulas and t is a term;*
- *a tuple of the form $\langle (RTC_{x,y}\, \varphi)(s,t), r, z, \Gamma, \Delta \rangle$ where $(RTC_{x,y}\, \varphi)(s,t)$ is a formula, r is a term, Γ and Δ are finite sequences of formulas, and z is a variable not occurring free in Γ, Δ, or $(RTC_{x,y}\, \varphi)(s,t)$; or*
- *a tuple of the form $\langle s = t, x, \varphi, n, \Gamma, \Delta \rangle$ where s and t are terms, x is a variable, φ is a formula, $n \in \{1, 2\}$, and Γ and Δ are finite sequences of formulas.*

A schedule *is a recursive enumeration of schedule elements in which every schedule element appears infinitely often (these exist since our language is countable).*

Each schedule corresponds to an exhaustive search strategy for a cut-free proof for each sequent $\Gamma \Rightarrow \Delta$, via the following notion of a 'search tree'.

Definition 15 (Search Tree). *Given a schedule $\{E_i\}_{i>0}$, for each sequent $\Gamma \Rightarrow \Delta$ we inductively define an infinite sequence of (possibly open) derivation trees, $\{T_i\}_{i>0}$, such that T_1 consists of the single open node $\Gamma \Rightarrow \Delta$, and each T_{i+1} is obtained by replacing all suitable open nodes in T_i with applications of first axioms and then the left and right inference rules for the formula in the ith schedule element. We show the cases for building T_{i+1} for when E_i corresponds to an RTC formula and an equality formula. The cases for when E_i corresponds to a standard compound first-order formula are similar.*

- *When E_i is of the form $\langle (RTC_{x,y}\, \varphi)(s,t), r, z, \Gamma, \Delta \rangle$, then T_{i+1} is obtained by:*

1. first closing as such any open node that is an instance of an axiom (after left and right weakening, if necessary);

2. next, replacing every open node $\Gamma', (RTC_{x,y}\, \varphi)(s,t) \Rightarrow \Delta'$ of the resulting tree for which $\Gamma' \subseteq \Gamma$ and $\Delta' \subseteq \Delta$ with the derivation:

$$\frac{\Gamma', (RTC_{x,y}\, \varphi)(s,t), s = t \Rightarrow \Delta' \qquad \Gamma', (RTC_{x,y}\, \varphi)(s,t), (RTC_{x,y}\, \varphi)(s,z), \varphi\left\{\frac{z}{x}, \frac{t}{y}\right\} \Rightarrow \Delta'}{\Gamma', (RTC_{x,y}\, \varphi)(s,t) \Rightarrow \Delta'} \ (4)$$

3. finally, replacing every open node $\Gamma' \Rightarrow \Delta', (RTC_{x,y}\, \varphi)(s,t)$ of the resulting tree with the derivation:

$$\frac{\Gamma' \Rightarrow \Delta', (RTC_{x,y}\, \varphi)(s,t), (RTC_{x,y}\, \varphi)(s,r) \qquad \Gamma' \Rightarrow \Delta', (RTC_{x,y}\, \varphi)(s,t), \varphi\left\{\frac{r}{x}, \frac{t}{y}\right\}}{\Gamma' \Rightarrow \Delta', (RTC_{x,y}\, \varphi)(s,t)} \ (2)$$

- *When E_i is of the form $\langle s = t, x, \varphi, n, \Gamma, \Delta \rangle$, then T_{i+1} is then obtained by first closing as such any open node that is an instance of an axiom (after*

left and right weakening, if necessary); and next, replacing every open node
$\Gamma', s = t \Rightarrow \Delta', \psi$ *in the resulting tree where* $\Gamma' \subseteq \Gamma$ *and* $\Delta' \subseteq \Delta$, *and* ψ *is* $\varphi\left\{\frac{s}{x}\right\}$
(resp. $\varphi\left\{\frac{t}{x}\right\}$*) if* $n = 1$ *(resp.* $n = 2$*), with the appropriate one of the following derivations:*

$$\frac{\Gamma', s = t \Rightarrow \Delta', \varphi\left\{\frac{t}{x}\right\}, \varphi\left\{\frac{s}{x}\right\}}{\Gamma', s = t \Rightarrow \Delta', \varphi\left\{\frac{t}{x}\right\}} \ (=L_1) \qquad \frac{\Gamma', s = t \Rightarrow \Delta', \varphi\left\{\frac{s}{x}\right\}, \varphi\left\{\frac{t}{x}\right\}}{\Gamma', s = t \Rightarrow \Delta', \varphi\left\{\frac{s}{x}\right\}} \ (=L_2)$$

The limit of the sequence $\{T_i\}_{i>0}$ *is a possibly infinite (and possibly open) derivation tree called the* search tree *for* $\Gamma \Rightarrow \Delta$ *with respect to the schedule* $\{E_i\}_{i>0}$, *and denoted by* T_ω.

Search trees are, by construction, recursive and cut-free. We construct special 'sequents' out of search trees, called *limit sequents*, as follows.

Definition 16 (Limit Sequents). *When a search tree* T_ω *is not an* RTC_G^ω *proof, either: (1) it is not even a pre-proof, i.e. it contains an open node; or (2) it is a pre-proof but contains an infinite branch that fails to satisfy the global trace condition. In case (1) it contains an open node to which, necessarily, no schedule element applies (e.g. a sequent containing only atomic formulas), for which we write* $\Gamma_\omega \Rightarrow \Delta_\omega$. *In case (2) the global trace condition fails, so there exists an infinite path* $\{\Gamma_i \Rightarrow \Delta_i\}_{i>0}$ *in* T_ω *which is followed by no infinitely progressing traces; we call this path the* untraceable branch *of* T_ω. *We then define* $\Gamma_\omega = \bigcup_{i>0} \Gamma_i$ *and* $\Delta_\omega = \bigcup_{i>0} \Delta_i$, *and call* $\Gamma_\omega \Rightarrow \Delta_\omega$ *the* limit sequent.[4]

Note that use of the word 'sequent' here is an abuse of nomenclature, since limit sequents may be infinite and thus technically not sequents. However when we say that such a limit sequent is provable, we mean that it has a finite subsequent that is provable.

Lemma 2. *Limit sequents* $\Gamma_\omega \Rightarrow \Delta_\omega$ *are not cut-free provable.*

Proof. Straightforward adaptation of the proof of [6, Lemma 6.3].

As standard, we use a limit sequent to induce a counter-interpretation, consisting of a Herbrand model quotiented by the equalities found in the limit sequent.

Definition 17 (Quotient Relation). *For a limit sequent* $\Gamma_\omega \Rightarrow \Delta_\omega$, *the relation* $\sim$ *is defined as the smallest congruence relation on terms such that* $s \sim t$ *whenever* $s = t \in \Gamma_\omega$. *We write* $[t]$ *for the* $\sim$-*equivalence class of* t, *i.e.* $[t] = \{u \mid t \sim u\}$.

The following property holds of the quotient relation.

Lemma 3. *If* $t \sim u$, *then* $\Gamma_\omega \Rightarrow F\left\{\frac{t}{x}\right\}$ *is cut-free provable in* RTC_G^ω *if and only if* $\Gamma_\omega \Rightarrow F\left\{\frac{u}{x}\right\}$.

---

[4] To be rigorous, we may pick e.g. the left-most open node or untraceable branch.

Proof. By induction on the conditions defining $\sim$. We use $\equiv$ to denote syntactic equality on terms, in order to distinguish from formulas $s = t$ asserting equality between (interpretations of) terms.

$(t \sim t)$: Immediate, since then $t \equiv u$.

$(t = u \in \Gamma_\omega)$: Assume $\Gamma_\omega \Rightarrow F\left\{\frac{t}{x}\right\}$ is cut-free provable, then we can apply the $(=L_1)$ rule to derive (without cut) $\Gamma_\omega, t = u \Rightarrow F\left\{\frac{u}{x}\right\}$; however notice that $\Gamma_\omega, t = u$ is simply Γ_ω since $t = u \in \Gamma_\omega$ already. The converse direction is symmetric, using rule $(=L_2)$.

$(t \sim u \Rightarrow u \sim t)$: Immediate, by induction.

$(t \sim u \land u \sim v \Rightarrow t \sim v)$: Straightforward, by induction.

$(t_1 \sim u_1 \land \ldots \land t_n \sim u_n \Rightarrow f(t_1, \ldots, t_n) \sim f(u_1, \ldots, u_n))$: Consider the formula F; clearly there exist formulas $G_1, \ldots, G_n$ and some variable y such that $G_i\left\{\frac{t}{y}\right\} \equiv F\left\{\frac{f(u_1, \ldots, u_{i-1}, t_i, \ldots, t_n)}{x}\right\}$ for each $i \leq n$. By induction, each sequent $\Gamma_\omega \Rightarrow G_i\left\{\frac{t_i}{y}\right\}$ is cut-free provable if and only if so too is $\Gamma_\omega \Rightarrow G_i\left\{\frac{u_i}{y}\right\}$. The result then follows since $F\left\{\frac{f(t_1, \ldots, t_n)}{x}\right\} \equiv G_1\left\{\frac{t_1}{y}\right\}$ and $F\left\{\frac{f(u_1, \ldots, u_n)}{x}\right\} \equiv G_n\left\{\frac{u_n}{y}\right\}$, and also $G_i\left\{\frac{u_i}{y}\right\} \equiv G_{i+1}\left\{\frac{t_{i+1}}{y}\right\}$ for each $i < n$.

We define the counter-interpretation as follows.

Definition 18 (Counter-interpretations). *Assume a search tree T_ω which is not a* RTC_G^ω *proof with limit sequent $\Gamma_\omega \Rightarrow \Delta_\omega$. Define a structure $M_\omega = \langle D, I \rangle$ as follows:*

- *$D = \{[t] \mid t$ is a term$\}$ (i.e. the set of terms quotiented by the relation $\sim$).*
- *For every k-ary function symbol f: $I(f)([t_1], \ldots, [t_k]) = [f(t_1, \ldots, t_k)]$*
- *For every k-ary relation symbol q: $I(q) = \{([t_1], \ldots, [t_k]) \mid q(t_1, \ldots, t_k) \in \Gamma_\omega\}$*

We also define a valuation ρ_ω for M_ω by $\rho_\omega(x) = [x]$ for all variables x.

The counter-interpretation $\langle M_\omega, \rho_\omega \rangle$ has the following property, meaning that M_ω is a counter-model for the corresponding sequent $\Gamma \Rightarrow \Delta$ if its search tree T_ω is not a proof.

Lemma 4. *If $\psi \in \Gamma_\omega$ then $M_\omega, \rho_\omega \models \psi$; and if $\psi \in \Delta_\omega$ then $M_\omega, \rho_\omega \not\models \psi$.*

Proof. By well-founded induction using the lexicographic ordering of the number of binders (i.e. $\exists$, $\forall$, and RTC) in ψ and the structure of ψ. Notice that, by definition, $\rho_\omega(t) = [t]$ for all terms t.

For ψ atomic (i.e. of the form $q(t_1, \ldots, t_k)$), if $\psi \in \Gamma_\omega$ then it follows immediately by Definition 18 that $M_\omega, \rho_\omega \models q(t_1, \ldots, t_k)$. If, on the other hand, $\psi \in \Delta_\omega$ then assume for contradiction that indeed $M_\omega, \rho_\omega \models q(t_1, \ldots, t_k)$. It then follows from Definition 18 that there are terms $u_1, \ldots, u_k$ such that $q(u_1, \ldots, u_k) \in \Gamma_\omega$ and $u_i \sim t_i$ for each $i \leq k$. Notice that then we can prove $\Gamma_\omega \Rightarrow q(u_1, \ldots, u_k)$ axiomatically, and so it follows by (k applications of) Lemma 3 that $\Gamma_\omega \Rightarrow q(t_1, \ldots, t_k)$ is cut-free provable. However, since $q(t_1, \ldots, t_k) \in \Delta_\omega$, this would mean that the limit sequent $\Gamma_\omega \Rightarrow \Delta_\omega$ is cut-free provable, which contradicts Lemma 2. Thus we conclude that in fact $M_\omega, \rho_\omega \not\models q(t_1, \ldots, t_k)$.

For ψ an equality formula $s = t$, if $\psi \in \Gamma_\omega$ then we have immediately by Definition 18 that $\rho_\omega(s) = \rho_\omega(t)$ and thus that $M_\omega, \rho_\omega \models s = t$ by Definition 2. If, on the other hand, $s = t \in \Delta_\omega$, suppose for contradiction that indeed $M_\omega, \rho_\omega \models s = t$. It then follows from Definition 18 that $s \sim t$. Since we may derive $\Gamma_\omega \Rightarrow s = s$ axiomatically, it thus follows from Lemma 3 that there is a cut-free proof of $\Gamma_\omega \Rightarrow s = t$. However, since $s = t \in \Delta_\omega$ this would mean that the limit sequent $\Gamma_\omega \Rightarrow \Delta_\omega$ is cut-free provable, which contradicts Lemma 2. We thus conclude that in fact $M_\omega, \rho_\omega \not\models s = t$.

The cases where ψ is a standard compound first-order formula follow easily by induction.

In case $\psi = (RTC_{x,y}\,\varphi)(s, t)$, we reason as follows.

$-$ For the first part of the lemma assume $(RTC_{x,y}\,\varphi)(s, t) \in \Gamma_\omega$. Then, by the construction of T_ω, there is at least one occurrence of rule (4) with active formula ψ in the untraceable branch; thus there a two cases:

i) The branch follows the left-hand premise, so there is $s = t \in \Gamma_\omega$. Therefore, by Definition 18, $\rho_\omega(s) = \rho_\omega(t)$ and so it follows immediately from Definition 2 that $M_\omega, \rho_\omega \models (RTC_{x,y}\,\varphi)(s, t)$.

ii) The branch follows the right-hand premise and, since there is no infinitely progressing trace along the untraceable branch, there must be a finite number of distinct variables $z_1, \ldots, z_n$ $(n > 0)$ such that $\varphi\left\{\frac{z_i}{x}, \frac{z_{i+1}}{y}\right\} \in \Gamma_\omega$, for each $i < n$, and $\varphi\left\{\frac{z_n}{x}, \frac{t}{y}\right\} \in \Gamma_\omega$. Then, by the I.H., $M_\omega, \rho_\omega \models \varphi\left\{\frac{z_i}{x}, \frac{z_{i+1}}{y}\right\}$ for each $i < n$, and $M_\omega, \rho_\omega \models \varphi\left\{\frac{z_n}{x}, \frac{t}{y}\right\}$. Thus, $M_\omega, \rho_\omega[x := [z_i], y := [z_{i+1}]] \models \varphi$ for each $i < n$, and $M_\omega, \rho_\omega[x := [z_n], y := [t]] \models \varphi$. Moreover, the untraceable branch also follows the left-hand premise of rule (4) with active formula $(RTC_{x,y}\,\varphi)(s, z_1)$. Thus $s = z_1 \in \Gamma_\omega$, and so $\rho_\omega(s) = \rho_\omega(z_1) = [z_1]$. We then have from Definition 2 that $M_\omega, \rho_\omega \models \psi$.

$-$ For the second part of the lemma we first prove, by an inner induction on n, the following auxiliary result for all terms s and t and elements $a_0, \ldots, a_n \in D$ $(n > 0)$:

if $(RTC_{x,y}\,\varphi)(s, t) \in \Delta_\omega$, with $\rho_\omega(s) = a_0$ and $\rho_\omega(t) = a_n$, then there exists some $i < n$ such that $M_\omega, \rho_\omega[x := a_i, y := a_{i+1}] \not\models \varphi$.

$(n = 1)$: Since $(RTC_{x,y}\,\varphi)(s, t) \in \Delta_\omega$, we have $\varphi\left\{\frac{s}{x}, \frac{t}{y}\right\} \in \Delta_\omega$ by construction as the untraceable branch must traverse an instance of rule (2) with $r \equiv s$ and moreover must traverse the right-hand premise (otherwise, we would have $(RTC_{x,y}\,\varphi)(s, s) \in \Delta_\omega$ resulting in the branch being closed by an instance of rule (1)). Thus by the outer induction it follows that $M_\omega, \rho_\omega \not\models \varphi\left\{\frac{s}{x}, \frac{t}{y}\right\}$ and thence that $M_\omega, \rho_\omega[x := \rho_\omega(s), y := \rho_\omega(t)] \not\models \varphi$ as required.

$(n = k + 1, \ k > 0)$: Then there exists some term r such that $a_k = [r] = \rho_\omega(r)$. If we have $(RTC_{x,y}\,\varphi)(s, t) \in \Delta_\omega$, by construction of the search tree T_ω we then also have that either $(RTC_{x,y}\,\varphi)(s, r) \in \Delta_\omega$ or $\varphi\left\{\frac{r}{x}, \frac{t}{y}\right\} \in \Delta_\omega$, as the untraceable branch must traverse an instance of rule (2) for the term r. In the

14

case of the former, the required result holds by the inner induction. In the case of the latter, we have $M_\omega, \rho_\omega \not\models \varphi \left\{ \frac{r}{x}, \frac{t}{y} \right\}$ by the outer induction and thence that $M_\omega, \rho_\omega[x := \rho_\omega(r), y := \rho_\omega(t)] \not\models \varphi$; i.e. $M_\omega, \rho_\omega[x := a_k, y := a_{k+1}] \not\models \varphi$ as required.

We now show that the primary result holds. Assume $(RTC_{x,y}\, \varphi)(s,t) \in \Delta_\omega$ and suppose for contradiction that $M_\omega, \rho_\omega \models (RTC_{x,y}\, \varphi)(s,t)$ holds. Thus, by Definition 2, there are two cases to consider.

- If $\rho_\omega(s) = \rho_\omega(t)$ then $s \sim t$. Thus since we may derive $\Gamma_\omega \Rightarrow (RTC_{x,y}\, \varphi)(s,s)$ by applying rule (1), by Lemma 3 there must also be a cut-free proof of $\Gamma_\omega \Rightarrow (RTC_{x,y}\, \varphi)(s,t)$. However, since $(RTC_{x,y}\, \varphi)(s,t) \in \Delta_\omega$ this would imply that $\Gamma_\omega \Rightarrow \Delta_\omega$ is cut-free provable, which contradicts Lemma 2.

- Otherwise, there are $a_0, \ldots, a_1 \in D$ $(n > 0)$ such that $\rho_\omega(s) = a_0$, $\rho_\omega(t) = a_n$ and $M_\omega, \rho_\omega[x := a_i, y := a_{i+1}] \models \varphi$ for each $i < n$. However this directly contradicts the auxiliary result proved above.

In both cases, we have derived a contradiction, and so we then conclude that $M_\omega, \rho_\omega \not\models (RTC_{x,y}\, \varphi)(s,t)$ as required. $\qquad\square$

The completeness result therefore follows since, by construction, a sequent S is contained within its corresponding limit sequents.

Theorem 3 (Completeness). RTC_G^ω *is complete for standard semantics.*

Proof. Now given any sequent S, if some search tree T_ω contracted for S is not an RTC_G^ω proof then it follows from Lemma 4 that S is not valid (M_ω is a counter model for it). Thus if S is valid, then T_ω is a recursive RTC_G^ω proof for it. $\qquad\square$

We obtain admissibility of cut as the search tree T_ω is cut-free.

Corollary 2 (Cut admissibility). *Cut is admissible in* RTC_G^ω.

3.4 $\mathcal{L}_{\mathsf{RTC}}$ with Pairs

To obtain the full inductive expressivity we must allow the formation of the transitive closure of not only binary relations, but any $2n$-ary relation. In [1] it was shown that taking such a RTC^n operator for every n (instead of just for $n = 1$) results in a more expressive logic, namely one that captures all finitary first-order definable inductive definitions and relations. Nonetheless, from a proof theoretical point of view having infinitely many such operators is suboptimal. Thus, we here instead incorporate the notion of ordered pairs and use it to encode such operators. For example, writing $\langle x, y \rangle$ for the application of the pairing function $\langle \rangle (x, y)$, the formula $(RTC^2_{x_1,x_2,y_1,y_2}\, \varphi)(s_1, s_2, t_1, t_2)$ can be encoded by:

$$(RTC_{x,y}\, \exists x_1, x_2, y_1, y_2\, .\, x = \langle x_1, x_2 \rangle \wedge y = \langle y_1, y_2 \rangle \wedge \varphi)(\langle s_1, s_2 \rangle, \langle t_1, t_2 \rangle)$$

Accordingly, we may assume languages that explicitly contain a pairing function, providing that we (axiomatically) restrict to structures that interpret it as such

15

(i.e. the *admissible* structures). For such languages we can consider two induced semantics: admissible standard semantics and admissible Henkin semantics, obtained by restricting the (first-order part of the) structures to be admissible.

The above proof systems are extended to capture ordered pairs as follows.

Definition 19. *For a signature containing at least one constant c, and a binary function symbol denoted by $\langle\rangle$, the proof systems $\langle\mathsf{RTC}\rangle_G$, $\langle\mathsf{RTC}\rangle_G^\omega$, and $\langle\mathsf{CRTC}\rangle_G^\omega$ are obtained from RTC_G, RTC_G^ω, CRTC_G^ω (respectively) by the addition of the following rules:*

$$\frac{\Gamma \Rightarrow \langle x,y\rangle = \langle u,v\rangle, \Delta}{\Gamma \Rightarrow x = u \wedge y = v, \Delta} \qquad\qquad \frac{}{\Gamma, \langle x,y\rangle = c \Rightarrow \Delta}$$

The proofs of Theorems 1 and 3 can easily be extended to obtain the following results for languages with a pairing function. For completeness, the key observation is that the model of the counter-interpretation is one in which every binary function is a pairing function. That is, the interpretation of any binary function is such that satisfies the standard pairing axioms. Therefore, the model of the counter-interpretation is an admissible structure.

Theorem 4 (Soundness and Completeness of $\langle\mathsf{RTC}\rangle_G$ and $\langle\mathsf{RTC}\rangle_G^\omega$). *The proof systems $\langle\mathsf{RTC}\rangle_G$ and $\langle\mathsf{RTC}\rangle_G^\omega$ are both sound and complete for the admissible forms of Henkin and standard semantics, respectively.*

4 Relating the Finitary and Infinitary Proof Systems

This section discusses the relation between the explicit and the cyclic system for TC. In Section 4.1 we show that the former is contained in the latter. The converse direction, which is much more subtle, is discussed in Section 4.2.

4.1 Inclusion of RTC_G in CRTC_G^ω

Provability in the explicit induction system implies provability in the cyclic system. The key property is that we can derive the explicit induction rule in the cyclic system, as shown in Figure 2.

Lemma 5. *Rule (3) is derivable in CRTC_G^ω.*

This leads to the following result (an analogue to [6, Thm. 7.6]).

Theorem 5. $\mathsf{CRTC}_G^\omega \supseteq \mathsf{RTC}_G$, *and is thus complete w.r.t. Henkin semantics.*

Proof. Let $\mathcal{P}$ be a proof in RTC_G and $\mathcal{P}'$ be the corresponding pre-proof in CRTC_G^ω obtained be replaing each instance of Rule (3) by the corresponding instance of the proof schema given in Lemma 5. We argue that $\mathcal{P}'$ is a valid CRTC_G^ω proof. Notice that the only cycles in $\mathcal{P}'$ are internal to the subproofs that simulate Rule (3). Thus any infinite path in $\mathcal{P}'$ must eventually end up

$$
\cfrac{
\cfrac{
\cfrac{\Gamma, \psi\left\{\tfrac{v}{x}\right\}, (RTC_{x,y}\,\varphi)(v,w) \Rightarrow \Delta, \psi\left\{\tfrac{w}{x}\right\}}
{\Gamma, \psi\left\{\tfrac{v}{x}\right\}, (RTC_{x,y}\,\varphi)(v,z) \Rightarrow \Delta, \psi\left\{\tfrac{z}{x}\right\}}\ \text{(Subst)}
\quad
\cfrac{\Gamma, \psi, \varphi \Rightarrow \Delta, \psi\left\{\tfrac{y}{x}\right\}}
{\Gamma, \psi\left\{\tfrac{z}{x}\right\}, \varphi\left\{\tfrac{z}{x}, \tfrac{w}{y}\right\} \Rightarrow \Delta, \psi\left\{\tfrac{w}{x}\right\}}\ \text{(Subst)}
}
{\Gamma, \psi\left\{\tfrac{v}{x}\right\}, (RTC_{x,y}\,\varphi)(v,z), \varphi\left\{\tfrac{z}{x}, \tfrac{w}{y}\right\} \Rightarrow \Delta, \psi\left\{\tfrac{w}{x}\right\}}\ \text{(Cut)}
\qquad
\cfrac{\cfrac{\Gamma, \psi\left\{\tfrac{v}{x}\right\} \Rightarrow \Delta, \psi\left\{\tfrac{v}{x}\right\}}{\Gamma, \psi\left\{\tfrac{v}{x}\right\}, v=w \Rightarrow \Delta, \psi\left\{\tfrac{w}{x}\right\}}\ \text{(=L}_1)}{}\ \text{(WL,WR,Ax)}
}
{\cfrac{\Gamma, \psi\left\{\tfrac{v}{x}\right\}, (RTC_{x,y}\,\varphi)(v,w) \Rightarrow \Delta, \psi\left\{\tfrac{w}{x}\right\}}
{\Gamma, \psi\left\{\tfrac{s}{x}\right\}, (RTC_{x,y}\,\varphi)(s,t) \Rightarrow \Delta, \psi\left\{\tfrac{t}{x}\right\}}\ \text{(Subst)}}\ (4)
$$

Fig. 2: CRTC_G^ω derivation simulating Rule (3). The variables v and w are fresh (i.e. do not occur free in Γ, Δ, φ, or ψ).

traversing one of these cycles infinitely often. Therefore, it suffices to show that there is an infinitely progressing trace following each such path. This is clearly the case since we can trace the active RTC formulas along these paths, which progress once each time around the cycle, across Rule (4). $\qquad\square$

Lemma 5 is the TC counterpart of [6, Lemma 7.5]. It is interesting to note that the simulation of the explicit LKID induction rule in the cyclic LKID system is rather complex since each predicate has a slightly different explicit induction rule, which depends on the particular productions defining it. Thus, the construction for the cyclic LKID system must take into account the possible forms of arbitrary productions. In contrast, CRTC_G^ω provides a single, uniform way to unfold an RTC formula: the construction given in Fig. 2 is the cyclic representation of the RTC operator semantics, with the variables v and w implicitly standing for arbitrary terms (that we subsequently substitute for).

This uniform syntactic translation of the explicit RTC_G induction rule into CRTC_G^ω allows us to *syntactically* identify a proper subset of cyclic proofs which is also complete w.r.t. Henkin semantics.[5] The criterion we use is based on the notion of *overlapping cycles*. Recall the definition of a *basic* cycle, which is a path in a (proof) graph starting and ending at the same point, but containing no other repeated nodes. We say that two distinct (i.e. not identical up to permutation) basic cycles *overlap* if they share any nodes in common, i.e. at some point they both traverse the same path in the graph. We say that a cyclic proof is *non-overlapping* whenever no two distinct basic cycles it contains overlap. The restriction to non-overlapping proofs has an advantage for automation, since one has only to search for cycles in one single branch.

Definition 20 (Normal Cyclic Proofs). *The* normal *cyclic proof system* NCRTC_G^ω *is the subsystem of* RTC_G^ω *comprising of all and only the non-overlapping cyclic proofs.*

---

[5] Note it is not clear that a similar complete structural restriction is possible for LKID.

17

The following theorem is immediate due to the fact that the translation of an RTC_G proof into CRTC_G^ω, using the construction shown in Figure 2, results in a proof with no overlapping cycles.

Theorem 6. $\mathsf{NCRTC}_G^\omega \supseteq \mathsf{RTC}_G$.

Henkin-completeness of the normal cyclic system then follows from Theorem 6 and Theorem 1.

4.2 Inclusions of CRTC_G^ω in RTC_G

This section addresses the question of whether the cyclic system is equivalent to the explicit one, or strictly stronger. In [6] it was conjectured that for the system with inductive definitions, LKID and CLKID^ω are equivalent. Later, it was shown that they are indeed equivalent when containing arithmetics [3,24]. We obtain a corresponding theorem in Section 4.2.1 for the TC systems. However, it was also shown in [2] that in the general case the cyclic system is stronger than the explicit one. We discuss the general case for TC and its subtleties in Section 4.2.2.

4.2.1 The Case of Arithmetics

Let $\mathcal{L}_{\mathsf{RTC}}$ be a language based on the signature $\{0, \mathsf{s}, +\}$. Let $\mathsf{RTC}_G + \mathsf{A}$ and $\mathsf{CRTC}_G^\omega + \mathsf{A}$ be the systems for $\mathcal{L}_{\mathsf{RTC}}$ obtained by adding to RTC_G and CRTC_G^ω, respectively, the standard axioms of PA together with the RTC-characterization of the natural numbers, i.e.:

   i) $\mathsf{s}\,x = 0 \Rightarrow$
   ii) $\mathsf{s}\,x = \mathsf{s}\,y \Rightarrow x = y$
   iii) $\Rightarrow x + 0 = x$
   iv) $\Rightarrow x + \mathsf{s}\,y = \mathsf{s}\,(x + y)$
   v) $\Rightarrow (RTC_{w,u}\,\mathsf{s}\,w = u)(0, x)$

Note that we do not need to assume multiplication explicitly in the signature, nor do we need to add axioms for it, since multiplication is definable in $\mathcal{L}_{\mathsf{RTC}}$ and its standard axioms are derivable [1,11].

Recall that we can express facts about sequences of numbers in PA by using a β-function such that for any finite sequence $k_0, k_1, ..., k_n$ there is some c such that for all $i \leq n$, $\beta(c, i) = k_i$. Accordingly, let B be a well-formed formula of the language of PA with three free variables which captures in PA a β-function. For each formula φ of the language of PA define $\varphi^\beta := \varphi$, and define $((RTC_{x,y}\,\varphi)(s,t))^\beta$ to be:

$$s = t \vee (\exists z, c \,.\, B(c, 0, s) \wedge B(c, \mathsf{s}\,z, t) \wedge$$
$$(\forall u \leq z \,.\, \exists v, w \,.\, B(c, u, v) \wedge B(c, \mathsf{s}\,u, w) \wedge \varphi^\beta \left\{ \tfrac{v}{x}, \tfrac{w}{y} \right\}))$$

The following result, which was proven in [8,11], establishes an equivalence between $\mathsf{RTC}_G + \mathsf{A}$ and PA_G (a Gentzen-style system for PA). It is mainly based on the fact that in $\mathsf{RTC}_G + \mathsf{A}$ all instances of PA_G induction rule are derivable.

Theorem 7 (cf. [11]). *The following hold:*

1. $\vdash_{\mathsf{RTC}_G+\mathsf{A}} \varphi \Leftrightarrow \varphi^\beta$.
2. $\vdash_{\mathsf{RTC}_G+\mathsf{A}} \Gamma \Rightarrow \Delta$ *iff* $\vdash_{\mathsf{PA}_G} \Gamma^\beta \Rightarrow \Delta^\beta$.

We show a similar equivalence holds between the cyclic system CRTC_G^ω and CA_G, a cyclic system for arithmetic shown to be equivalent to PA_G [24]. The first part of the result is straightforward.

Lemma 6. $\vdash_{\mathsf{CRTC}_G^\omega+\mathsf{A}} \varphi \Leftrightarrow \varphi^\beta$

Proof. This follows from Theorem 7(1) and Theorem 5. □

To show the second part, we first show that CA_G is included in $\mathsf{CRTC}_G^\omega+\mathsf{A}$ by giving a construction that directly translates CA_G proofs into $\mathsf{CRTC}_G^\omega+\mathsf{A}$ proofs. Technically, the signature of CA_G includes the relation symbol $<$ for strict ordering, and the function symbol $\cdot$ for multiplication. As mentioned above, multiplication (and its axioms) are derivable in TC, and the non-strict ordering on natural numbers $s \leq t$ can be expressed in TC as $(RTC_{w,u}\,\mathsf{s}\,w = u)(s,t)$. Therefore, in the following result, we implicitly assume that all CA_G terms of the form $s \cdot t$ are translated in CRTC_G^ω as in [1,11], and formulas of the form $s < t$ as $s \neq t \wedge (RTC_{w,u}\,\mathsf{s}\,w = u)(s,t)$.

Lemma 7. *If* $\vdash_{\mathsf{CA}_G} \Gamma \Rightarrow \Delta$ *then* $\vdash_{\mathsf{CRTC}_G^\omega+\mathsf{A}} \Gamma \Rightarrow \Delta$.

Proof. We define a translation $[\cdot]^*$ on sets of formulas Γ as the smallest set of formulas containing Γ and $(RTC_{w,u}\,\mathsf{s}\,w = u)(0,t)$ for each term t that appears as a subterm of a formula in Γ with no quantified variables (we call such a term *free*). Notice it suffices to prove $\vdash_{\mathsf{CRTC}_G^\omega+\mathsf{A}} \Gamma^* \Rightarrow \Delta$ if $\vdash_{\mathsf{CA}_G} \Gamma \Rightarrow \Delta$, since then we may cut all the added formulas $(RTC_{w,u}\,\mathsf{s}\,w = u)(0,t)$ in Γ^* using instances of axiom (v).

We call $\Gamma^* \Rightarrow \Delta$ the $*$-translation of the sequent $\Gamma \Rightarrow \Delta$, and call the inference rule obtained by applying this translation to the conclusion and each premise the $*$-translation of the rule. We begin by showing that the $*$-translations of the axioms and proof rules of CA_G are derivable in $\mathsf{CRTC}_G^\omega+\mathsf{A}$. For the standard rules of $\mathcal{LK}$ and the substitution rule this is trivial, and can be done is such a way that, for the conclusion $\Gamma \Rightarrow \Delta$ and each premise $\Gamma' \Rightarrow \Delta'$, there is a (non-progressing) trace from each $(RTC_{w,u}\,\mathsf{s}\,w = u)(0,t) \in \Gamma^*$ to $(RTC_{w,u}\,\mathsf{s}\,w = u)(0,t') \in (\Gamma')^*$ when t' is a predecessor of t (cf. [24, Def. 1]). It remains to show that we can derive the $*$-translations of the following set of axioms:

$$t < u, u < v \Rightarrow t < v$$
$$t < u, u < t \Rightarrow$$
$$t < u, u < \mathsf{s}\,t \Rightarrow$$
$$t < 0 \Rightarrow$$
$$t < u \Rightarrow \mathsf{s}\,t < \mathsf{s}\,u$$
$$\Rightarrow t < \mathsf{s}\,t$$

$$\Rightarrow t < u, t = u, u < t$$
$$\Rightarrow t + 0 = t$$
$$\Rightarrow t + \mathsf{s}\,u = \mathsf{s}\,(t + u)$$
$$\Rightarrow t \cdot 0 = 0$$
$$\Rightarrow t \cdot \mathsf{s}\,u = (t \cdot u) + t$$

$$\Rightarrow z < t', t' = z, t' < z \qquad \mathsf{N}\,t', \mathsf{N}\,\mathsf{s}\,z, z < t', t' < \mathsf{s}\,z \Rightarrow \quad \text{(Cut)}$$
$$\frac{\mathsf{N}\,t', \mathsf{N}\,\mathsf{s}\,z, t' < \mathsf{s}\,z \Rightarrow t' = z, t' < z}{\mathsf{N}\,t', \mathsf{N}\,\mathsf{s}\,z, t' < \mathsf{s}\,z \Rightarrow t' = z \lor t' < z} \ (\lor\text{R})$$

$$\frac{t' < 0, \mathsf{N}\,t', \mathsf{N}\,0 \Rightarrow}{\Gamma, t' < t, t' < 0, \mathsf{N}\,t', \mathsf{N}\,t, \mathsf{N}\,0 \Rightarrow \Delta} \ (\text{WL/WR})$$

$$\frac{\dfrac{\dfrac{\Gamma, t' < t, t' < \mathsf{s}\,z, \mathsf{N}\,t', \mathsf{N}\,t, \mathsf{N}\,\mathsf{s}\,z, \mathsf{N}\,z \Rightarrow \Delta}{\Gamma, t' < t, t' < \mathsf{s}\,z', \mathsf{N}\,t', \mathsf{N}\,t, \mathsf{N}\,\mathsf{s}\,z', \mathsf{N}\,z' \Rightarrow \Delta} \ (\text{Subst})}{\Gamma, t' < t, t' < z, \mathsf{N}\,t', \mathsf{N}\,t, \mathsf{N}\,z', z = \mathsf{s}\,z' \Rightarrow \Delta} \ (\text{=L}')}{\Gamma, t' < t, t' < z, \mathsf{N}\,t', \mathsf{N}\,t, \mathsf{N}\,z \Rightarrow \Delta} \ (4)$$

$$\frac{* \quad \dfrac{\Gamma, t' < t, \mathsf{N}\,t', \mathsf{N}\,t \Rightarrow \Delta}{\Gamma, t' < t, t' = z, \mathsf{N}\,t', \mathsf{N}\,t, \mathsf{N}\,z \Rightarrow \Delta} \ (\text{WL})}{\dfrac{\Gamma, t' < t, t' = z \lor t' < z, \mathsf{N}\,t', \mathsf{N}\,t, \mathsf{N}\,z \Rightarrow \Delta}{\Gamma, t' < t, t' < \mathsf{s}\,z, \mathsf{N}\,t', \mathsf{N}\,t, \mathsf{N}\,\mathsf{s}\,z, \mathsf{N}\,z \Rightarrow \Delta}} \ (\lor\text{L}) \ (\text{Cut})$$

$$\frac{\dfrac{t' < 0, \mathsf{N}\,t', \mathsf{N}\,0 \Rightarrow}{\Gamma, t' < 0, \mathsf{N}\,t', \mathsf{N}\,0 \Rightarrow \Delta} \ (\text{WL/WR}) \qquad \dfrac{\Gamma, t' < t, \mathsf{N}\,t', \mathsf{N}\,t, \mathsf{N}\,z, t = \mathsf{s}\,z \Rightarrow \Delta}{} \ \begin{array}{l}(\text{=L}')\\(4)\end{array}}{\Gamma, t' < t, \mathsf{N}\,t', \mathsf{N}\,t \Rightarrow \Delta}$$

Fig. 3: A derivation schema simulating a CA_G trace progression point in $\mathsf{CRTC}_G^\omega + \mathsf{A}$.

which is straightforward. It is also similarly straightforward to show that the $*$-translation of the following inference rule:

$$\frac{\Gamma, t = \mathsf{s}\,x \Rightarrow \Delta}{\Gamma, 0 < t \Rightarrow \Delta} \ (x \text{ is fresh})$$

is derivable in $\mathsf{CRTC}_G^\omega + \mathsf{A}$ such that there is a (non-progressing) trace from each $(RTC_{w,u}\,\mathsf{s}\,w = u)(0,t) \in \Gamma^*$ in the conclusion to the corresponding formula in the premise of the derivation. We shall call these derivations of the $*$-translations of the CA_G axioms ad rules the *simple* derivations.

We next show that for each (non-axiomatic) rule of CA_G, we can derive the $*$-translation of the rule in $\mathsf{CRTC}_G^\omega + \mathsf{A}$ with the additional property that for every *progressing* trace from a term t in the conclusion to a term t' in a premise, there is also a progressing trace from the formula $(RTC_{w,u}\,u = \mathsf{s}\,w)(0,t)$ in the conclusion of the derived rule to the formula $(RTC_{w,u}\,u = \mathsf{s}\,w)(0,t')$ in the corresponding premise.

Consider the open derivation schema shown in Figure 3. Here, $\mathsf{N}\,t$ abbreviates the formula $(RTC_{w,u}\,u = \mathsf{s}\,w)(0,t)$, and $t' < t$ abbreviates the translation given above, i.e. $s \neq t \land (RTC_{w,u}\,\mathsf{s}\,w = u)(s,t)$. The symbols †, ‡ and ✠ denote the

(simple) derivations of the $*$-translations of the appropriate axioms. We also write $(\mathrm{=L}')$ in Figure 3 to refer to instances of the following general schema for a derived equality rule, in which $\Sigma(t) = \{\varphi_1(t), \ldots, \varphi_n(t)\}$ and $\Sigma(t)$ may or may not occur in the right-hand premise of the (Cut) rule instance.

$$
\cfrac{
\cfrac{
\cfrac{
\cfrac{\varphi_1(t) \Rightarrow \varphi_1(t)}{\Sigma(t) \Rightarrow \varphi_1(t)} (\mathrm{WL}) \quad \cdots \quad \cfrac{\varphi_n(t) \Rightarrow \varphi_n(t)}{\Sigma(t) \Rightarrow \varphi_n(t)} (\mathrm{WL})
}{\Sigma(t) \Rightarrow \wedge_{i \le n} \varphi_i(t)} (\wedge\mathrm{R})
}{\Sigma(t), t = u \Rightarrow \wedge_{i \le n} \varphi_i(u)} (\mathrm{=L_1}) \quad
\cfrac{\Gamma, [\Sigma(t)], \Sigma(u) \Rightarrow \Delta}{\Gamma, [\Sigma(t)], \wedge_{i \le n} \varphi_i(u) \Rightarrow \Delta} (\wedge\mathrm{L})
}{\Gamma, \Sigma(t), t = u \Rightarrow \Delta} (\mathrm{Cut})
$$

Notice that, crucially, in the derivation in Figure 3 there is a *progressing* trace from $\mathsf{N}\,t$ in the conclusion to $\mathsf{N}\,t'$ in the premise (marked with a $*$). There is also a non-progressing trace from each RTC formula in Γ in the conclusion to its occurrence in Γ in the premise.

For each CA_G inference rule concluding $\Gamma \Rightarrow \Delta$, we build a *trace-aware* derivation of its $*$-translation as follows. First, we take the simple derivation of the inference rule. Notice that this gives a non-progressing trace from each $\mathsf{N}\,t \in \Gamma^*$ to $\mathsf{N}\,t'$ in a premise where t' is a precursor of t in the CA_G inference rule. Now, for every premise $\Gamma_j \Rightarrow \Delta_j$ of the rule, let $\{t'_1 < t_1, \ldots, t'_n < t_n\}$ be the set of all such formulas in Γ_j. Notice that $t'_i < t_i, \mathsf{N}\,t'_i, \mathsf{N}\,t_i \subseteq (\Gamma_j)^*$ for each i. To each corresponding premise in the simple derivation we apply, in turn, n instances of the derivation schema in Figure 3, one for each formula $t_i < t'_i$. Notice that this combined derivation satisfies the following for each premise $\Gamma_j \Rightarrow \Delta_j$ and free terms $t \in \Gamma$ and $t_i \in \Gamma_j$:

1. there is a non-progressing trace from $\mathsf{N}\,t \in \Gamma^*$ in the conclusion to $\mathsf{N}\,t_i \in (\Gamma_j)^*$ in the premise if t_i is a precursor of t; and
2. there is a progressing trace from $\mathsf{N}\,t \in \Gamma^*$ in the conclusion to $\mathsf{N}\,t'_i \in (\Gamma_j)^*$ in the premise if t_i is a precursor of t and $t'_i < t_i \in \Gamma_j$.

Note that this means the notion of trace in the CA_G inference rules is exactly mirrored by a CRTC_G^ω trace in the trace-aware derivation. Moreover, the properties above hold of *all* paths from the conclusion to a premise in the trace-aware derivation, in particular those that travel around the internal cycles any finite number of times.

From this construction it follows that we can transform a CA_G pre-proof, via the trace-aware local $*$-translation of each rule, into a $\mathsf{CRTC}_G^\omega + \mathsf{A}$ pre-proof with the same global structure. It remains to show that each such $\mathsf{CRTC}_G^\omega + \mathsf{A}$ pre-proof resulting from a CA_G proof is also a $\mathsf{CRTC}_G^\omega + \mathsf{A}$ proof. That is, it satisfies the CRTC_G^ω global trace condition. Consider an arbitrary infinite path in the $\mathsf{CRTC}_G^\omega + \mathsf{A}$ pre-proof. There are two cases to consider:

– The infinite path ends up traversing an infinite path local to the (trace-aware) $*$-translation of an inference rule or CA_G axiom; in this case notice that each such infinite path has an infinitely progressing trace.
– The infinite path corresponds to an infinite path in the CA_G proof (possibly interspersed with finite traversals of the cycles local to the trace-aware

∗-translation of each rule instance). Since there is an infinitely progressing trace following the path in the CA_G proof, by the properties above there is also a corresponding infinitely progressing trace following the path in the $\mathsf{CRTC}_G^\omega{+}\mathsf{A}$ pre-proof. □

This leads immediately to the 'if' direction.

Corollary 3. *If* $\vdash_{\mathsf{CA}_G} \Gamma^\beta \Rightarrow \Delta^\beta$ *then* $\vdash_{\mathsf{CRTC}_G^\omega{+}\mathsf{A}} \Gamma \Rightarrow \Delta$.

Proof. We first use Lemma 7 to derive $\Gamma^\beta \Rightarrow \Delta^\beta$ in $\mathsf{CRTC}_G^\omega{+}\mathsf{A}$, and then combine this with derivations in $\mathsf{CRTC}_G^\omega{+}\mathsf{A}$ of $\varphi \Rightarrow \varphi^\beta$ (resp. $\varphi^\beta \Rightarrow \varphi$) for each $\varphi \in \Gamma$ (resp. $\varphi \in \Delta$), which exist by Lemma 6, with applications of cuts to derive $\Gamma \Rightarrow \Delta$. □

In [3], to show the equivalence of the explicit and cyclic systems for LKID, a construction was given which translates LKID predicates $P(\boldsymbol{t})$ into predicates $P'(\boldsymbol{t}, n)$ with equivalent inductive definitions and an extra parameter n comprising a 'stage' variable. The equivalence is derived by using the cycles in a proof to construct an explicit induction hypothesis over these stage variables. Here, for TC, instead of directly constructing an induction hypothesis for the explicit system, we show that from a $\mathsf{CRTC}_G^\omega{+}\mathsf{A}$ proof we can construct an analogous proof in CA_G which preserves cycles, and then use the existing equivalence results between CA_G, PA_G and RTC_G. Our construction is similar to the one given in [3], in that we use a variant of the β-function which introduces a free variable n (similar to a stage variable), which we are able to trace in the cyclic CA_G proof. This results in the 'only if' direction of the result.

Lemma 8. *If* $\vdash_{\mathsf{CRTC}_G^\omega{+}\mathsf{A}} \Gamma \Rightarrow \Delta$ *then* $\vdash_{\mathsf{CA}_G} \Gamma^\beta \Rightarrow \Delta^\beta$.

Proof. Similarly to the proof of Lemma 7 above, we define a local translation on proof rules that preserves $\mathsf{CRTC}_G^\omega{+}\mathsf{A}$ traces as CA_G traces. For this, we use a *parameterised* variant $\bar{\beta}[n]$ of the β-translation, which introduces its parameter as a free variable in the translation. It is defined in the same way as the β-translation, except that it translates formulas of the form $(RTC_{x,y}\, \varphi)(s,t)$ as follows:

$$s = t \vee \exists z, c \,.\, n = \mathsf{s}\, z \wedge B(c, 0, s) \wedge B(c, \mathsf{s}\, z, t) \wedge$$
$$\forall u \leq z \,.\, \exists v, w \,.\, B(c, u, v) \wedge B(c, \mathsf{s}\, u, w) \wedge \varphi^\beta \left\{ \tfrac{v}{x}, \tfrac{w}{y} \right\}$$

Notice the use of the original β-translation for the body of the RTC formula φ. We extend the $\bar{\beta}$-translation to sets of formulas, sequents, and inference rules as follows:

– for a set of formulas Γ, we define $\Gamma^{\bar{\beta}}$ as the set of $\bar{\beta}$-translations of the formulas in Γ such that each translation of an RTC sub-formula introduces a fresh free variable z; that is, each distinct RTC sub-formula is translated using a distinct variable parameter;

22

– for sequents, we define $(\Gamma \Rightarrow \Delta)^{\bar{\beta}} = \Gamma^{\bar{\beta}} \Rightarrow \Delta^{\beta}$ such that the free variable parameters used by the $\bar{\beta}$-translation of the antecedent Γ are distinct from the free variables in the succedent Δ;

– for an inference rule with premises $\Gamma_1 \Rightarrow \Delta_1, \ldots, \Gamma_n \Rightarrow \Delta_n$ and conclusion $\Gamma \Rightarrow \Delta$, we define its $\bar{\beta}$-translation as the inference rule with premises $(\Gamma_1 \Rightarrow \Delta_1)^{\bar{\beta}}, \ldots, (\Gamma_n \Rightarrow \Delta_n)^{\bar{\beta}}$ and conclusion $(\Gamma \Rightarrow \Delta)^{\bar{\beta}}$ such that multiple occurrences of the same RTC sub-formula across the original premises and conclusion are translated using the same free variable parameter in each of the translated premises and conclusion.

We show that if $\vdash_{\mathsf{CRTC}_G^\omega + \mathsf{A}} \Gamma \Rightarrow \Delta$ then $\vdash_{\mathsf{CA}_G} (\Gamma \Rightarrow \Delta)^{\bar{\beta}}$. We first prove that the $\bar{\beta}$-translation of each $\mathsf{CRTC}_G^\omega + \mathsf{A}$ inference rule can be derived in CA_G in such a way that there is a CA_G trace (progressing or non-progressing, as appropriate) simulating each $\mathsf{CRTC}_G^\omega + \mathsf{A}$ trace present in the original rule, which we do by tracing the free variable parameters in the CA_G rule. We show how this is done for Rule (4); the other rules are straightforward, and do not contain progressing traces.

Take an instance of Rule (4) with contexts Γ and Δ, and active formulas $(RTC_{x,y}\,\varphi)(s,t)$ and $(RTC_{x,y}\,\varphi)(s,z)$ in the conclusion and right-hand premise, respectively. For terms r, s, and t, let:

– $\vartheta(r)$ abbreviate $\exists v,w\,.\,B(c,r,v) \wedge B(c,\mathsf{s}\,r,w) \wedge \varphi^\beta \left\{ \frac{v}{x},\frac{w}{y} \right\}$; and
– $A(r,s,t)$ abbreviate $\exists z,c\,.\,r = \mathsf{s}\,z \wedge B(c,0,s) \wedge B(c,\mathsf{s}\,z,t) \wedge \forall u \le z\,.\,\vartheta(u)$.

Additionally, let $\Sigma(r)$ and $\Pi(r)$ abbreviate the follow sequences of formulas:

$$n = \mathsf{s}\,r, B(c,0,s), B(c,\mathsf{s}\,r,t), \forall u \le r\,.\,\vartheta(u)$$

$$n = \mathsf{s}\,\mathsf{s}\,r, B(c,0,s), B(c,\mathsf{s}\,\mathsf{s}\,r,t), B(c,\mathsf{s}\,r,v), B(c,\mathsf{s}\,\mathsf{s}\,r,w), \varphi^\beta \left\{ \frac{v}{x},\frac{w}{y} \right\}, \forall u \le r\,.\,\vartheta(u)$$

Moreover, note the following.

i) We can easily derive $\Rightarrow z = 0 \vee \exists z'\,.\,z = \mathsf{s}\,z'$ using standard first-order rules and the axioms of CA_G; we refer to this derivation using †.

ii) Our use of the notation $\forall u \le t\,.\,\gamma$ technically abbreviates the CA_G formula $\forall u\,.\,(u = t \vee u < t) \to \gamma$, and so we may straightforwardly derive both $\forall u \le 0\,.\,\gamma(u) \Rightarrow \gamma(0)$ and $\forall u \le \mathsf{s}\,t\,.\,\gamma(u) \Rightarrow \gamma(\mathsf{s}\,t) \wedge \forall u \le t\,.\,\gamma(u)$; for brevity, we refer to an instance of the (Cut) rule that applies these sequents using the labels $(\le_0)$ and $(\le_\mathsf{s})$, respectively.

iii) Recall that, since the formula B captures a β-function, we may also derive $B(r,s,t), B(r,s,u) \Rightarrow t = u$; we abbreviate instances of (Cut) that apply an instance of this sequent using the label (B).

Using these elements, Fig. 4 shows a derivation of the following sequent, in which we have abbreviated the antecedent formula by ψ:

$$A(n,s,t) \Rightarrow \exists z,m\,.\,m < n \wedge (s = z \vee A(m,s,z)) \wedge \varphi^\beta \left\{ \frac{z}{x},\frac{t}{y} \right\}$$

Then, using Fig. 4 as a subderivation, we derive the $\bar{\beta}$-translation of Rule (4) in CA_G as shown in Fig. 5. Note that for any sequence of formulas Σ, we can straightforwardly derive $\Gamma, \Sigma^\beta \Rightarrow \Delta$ from $\Gamma, \Sigma^{\bar{\beta}} \Rightarrow \Delta$ by first introducing

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{\varphi^\beta\left\{\frac{s}{x},\frac{t}{y}\right\} \Rightarrow \varphi^\beta\left\{\frac{s}{x},\frac{t}{y}\right\}}\ (\mathsf{Ax})}{s=v,t=w,\varphi^\beta\left\{\frac{v}{x},\frac{w}{y}\right\} \Rightarrow \varphi^\beta\left\{\frac{s}{x},\frac{t}{y}\right\}}\ (=\!\mathsf{L})}{B(c,0,s),B(c,\mathsf{s}0,t),B(c,0,v),B(c,\mathsf{s}0,w),\varphi^\beta\left\{\frac{v}{x},\frac{w}{y}\right\}\Rightarrow\varphi^\beta\left\{\frac{z}{x},\frac{t}{y}\right\}}\ (B)}{B(c,0,s),B(c,\mathsf{s}0,t),\forall u\le 0\,.\,\vartheta(u)\Rightarrow\varphi^\beta\left\{\frac{z}{x},\frac{t}{y}\right\}}\ (\le_0/\exists\mathsf{L})}{z=0,\Sigma(z)\Rightarrow\varphi^\beta\left\{\frac{z}{x},\frac{t}{y}\right\}}}\ (\mathsf{WL}/=\!\mathsf{L})
$$

$$
\dfrac{\dfrac{\overline{z=0,\Sigma(z)\Rightarrow s=s}\ (=\!\mathsf{R})}{z=0,\Sigma(z)\Rightarrow s=s\vee A(0,s,s)}\ (\vee\mathsf{R})}{}
$$

$$
\dfrac{\dfrac{\overline{z=0,\Sigma(z)\Rightarrow 0<\mathsf{s}0}\ (\mathsf{PA\text{-}Ax})}{z=0,\Sigma(z)\Rightarrow 0<n}\ (=\!\mathsf{L})}{z=0,\Sigma(z)\Rightarrow\exists z,m\,.\,m<n\wedge(s=z\vee A(m,s,z))\wedge\varphi^\beta\left\{\frac{z}{x},\frac{t}{y}\right\}}\ (\exists\mathsf{R}/\wedge\mathsf{R})
$$

(a) One step from s to t.

$$
\dfrac{\dfrac{\overline{\Pi(z')\Rightarrow \mathsf{s}\,z'=\mathsf{s}\,z'}\ (=\!\mathsf{R})\quad \overline{\Pi(z')\Rightarrow B(c,0,s)}\ (\mathsf{Ax})\quad \overline{\Pi(z')\Rightarrow B(c,\mathsf{s}\,z',v)}\ (\mathsf{Ax})\quad \overline{\Pi(z')\Rightarrow\forall u\le z'\,.\,\vartheta(u)}\ (\mathsf{Ax})}{\Pi(z')\Rightarrow\exists z,c\,.\,\mathsf{s}\,z'=\mathsf{s}\,z\wedge B(c,0,s)\wedge B(c,\mathsf{s}\,z,v)\wedge\forall u\le z\,.\,\vartheta(u)}\ (\exists\mathsf{R}/\wedge\mathsf{R})}{\Pi(z')\Rightarrow s=v\vee A(\mathsf{s}\,z',s,v)}\ (\vee\mathsf{R})
$$

$$
\dfrac{\overline{\Pi(z')\Rightarrow\mathsf{s}\,z'<\mathsf{s}\,\mathsf{s}\,z'}\ (\mathsf{PA\text{-}Ax})}{\Pi(z')\Rightarrow\mathsf{s}\,z'<n}\ (=\!\mathsf{L})
$$

$$
\dfrac{\dfrac{\overline{\varphi^\beta\left\{\frac{v}{x},\frac{t}{y}\right\}\Rightarrow\varphi^\beta\left\{\frac{v}{x},\frac{t}{y}\right\}}\ (\mathsf{Ax})}{t=w,\varphi^\beta\left\{\frac{v}{x},\frac{w}{y}\right\}\Rightarrow\varphi^\beta\left\{\frac{v}{x},\frac{t}{y}\right\}}\ (=\!\mathsf{L})}{\Pi(z')\Rightarrow\varphi^\beta\left\{\frac{v}{x},\frac{t}{y}\right\}}\ (B/\mathsf{WL})
$$

$$
\dfrac{\dfrac{\dfrac{\Pi(z')\Rightarrow\exists z,m\,.\,m<n\wedge(s=z\vee A(m,s,z))\wedge\varphi^\beta\left\{\frac{z}{x},\frac{t}{y}\right\}}{n=\mathsf{s}\,\mathsf{s}\,z',B(c,0,s),B(c,\mathsf{s}\,\mathsf{s}\,z',t),\vartheta(\mathsf{s}\,z'),\forall u\le z'\,.\,\vartheta(u)\Rightarrow\psi}\ (\exists\mathsf{R}/\wedge\mathsf{R})}{n=\mathsf{s}\,\mathsf{s}\,z',B(c,0,s),B(c,\mathsf{s}\,\mathsf{s}\,z',t),\forall u\le\mathsf{s}\,z'\,.\,\vartheta(u)\Rightarrow\psi}\ (\exists\mathsf{L}/\wedge\mathsf{L})}{\exists z'\,.\,z=\mathsf{s}\,z',\Sigma(z)\Rightarrow\psi}\ (\exists\mathsf{L}/=\!\mathsf{L})
$$
$(\le_\mathsf{s})$

(b) Multi-step from s to t.

$$
\dfrac{\dfrac{\dfrac{\vdots\ \dagger}{\Rightarrow z=0\vee\exists z'\,.\,z=\mathsf{s}\,z'}\quad\dfrac{\overset{a}{\vdots}\quad\overset{b}{\vdots}}{\dfrac{z=0,\Sigma(z)\Rightarrow\psi\quad \exists z'\,.\,z=\mathsf{s}\,z',\Sigma(z)\Rightarrow\psi}{z=0\vee\exists z'\,.\,z=\mathsf{s}\,z',\Sigma(z)\Rightarrow\psi}\ (\vee\mathsf{L})}}{\Sigma(z)\Rightarrow\exists z,m\,.\,m<n\wedge(s=z\vee A(m,s,z))\wedge\varphi^\beta\left\{\frac{z}{x},\frac{t}{y}\right\}}\ (\mathsf{Cut})}{A(n,s,t)\Rightarrow\exists z,m\,.\,m<n\wedge(s=z\vee A(m,s,z))\wedge\varphi^\beta\left\{\frac{z}{x},\frac{t}{y}\right\}}\ (\exists\mathsf{L})
$$

Fig. 4: The core subderivation of the simulation of Rule (4) in CA_G.

existential quantifiers for the free variable parameters in $\Sigma^{\bar\beta}$ and then eliminating the terms $n<\mathsf{s}\,z$ with cuts. We abbreviate such a derivation using the label $(\bar\beta)$. Note that this admits non-progressing traces for all the free variable parameters in Γ. The crucial feature of this derivation is that there is a CA_G trace from the free variable parameter n in the conclusion to m in the right-hand premise, which progresses at the sequent containing the boxed formula $m<n$. Also, since

Fig. 4

$$\vdots$$

$$A(n,s,t) \Rightarrow \exists z, m \,.\, m < n \wedge (s = z \vee A(m,s,z)) \wedge \varphi^\beta \left\{ \tfrac{z}{x}, \tfrac{t}{y} \right\}$$

$$
\cfrac{
  \cfrac{
    \cfrac{
      \cfrac{
        \cfrac{
          \Gamma^{\bar\beta}, (s = z \vee A(m,s,z)), \varphi^\beta \left\{ \tfrac{z}{x}, \tfrac{t}{y} \right\} \Rightarrow \Delta^\beta
        }{
          \Gamma^{\bar\beta}, (s = z \vee A(m,s,z)), \varphi^\beta \left\{ \tfrac{z}{x}, \tfrac{t}{y} \right\} \Rightarrow \Delta^\beta
        } \; (\bar\beta)
      }{
        \Gamma^{\bar\beta}, \boxed{m < n}, (s = z \vee A(m,s,z)), \varphi^\beta \left\{ \tfrac{z}{x}, \tfrac{t}{y} \right\} \Rightarrow \Delta^\beta
      } \; \text{(WL)}
    }{
      \Gamma^{\bar\beta}, \exists z, m \,.\, m < n \wedge (s = z \vee A(m,s,z)) \wedge \varphi^\beta \left\{ \tfrac{z}{x}, \tfrac{t}{y} \right\} \Rightarrow \Delta^\beta
    } \; (\exists \text{L}/\wedge \text{L})
  }{
    \Gamma^{\bar\beta}, A(n,s,t) \Rightarrow \Delta^\beta
  } \; \text{(Cut)}
}{}
$$

$$\vdots$$

$$
\cfrac{
  \cfrac{
    (\Gamma')^{\bar\beta} \left\{ \tfrac{s}{u}, \tfrac{t}{w} \right\} \Rightarrow (\Delta')^\beta \left\{ \tfrac{s}{u}, \tfrac{t}{w} \right\}
  }{
    \Gamma^{\bar\beta}, s = t \Rightarrow \Delta^\beta
  } \; \text{(=L)}
  \qquad
  \cfrac{\vdots}{\Gamma^{\bar\beta}, A(n,s,t) \Rightarrow \Delta^\beta}
}{
  \Gamma^{\bar\beta}, s = t \vee A(n,s,t) \Rightarrow \Delta^\beta
} \; (\vee \text{L})
$$

Fig. 5: A derivation schema simulating Rule (4) in CA_G.

the context $\Gamma^{\bar\beta}$ is preserved along the paths to both the left and right premises, all non-progressing traces are simulated as well.

Now, using the derivations of the local $\bar\beta$-translations of the inference rules, from a CRTC_G^ω+A pre-proof, we can build a CA_G pre-proof with the same global structure. For each bud in the resulting CA_G pre-proof, we first apply an instance of the substitution rule that substitutes each free variable parameter of its companion with the free variable parameter of its corresponding $\bar\beta$-translation instance. Notice that this is possible, since the parameter variable is unique for the $\bar\beta$-translation of each RTC sub-formula. We can then form a cycle in the CA_G pre-proof.

Since the CRTC_G^ω traces for each rule are simulated by the CA_G derived rules, for each trace following a (finite or infinite) path in the CRTC_G^ω+A pre-proof there is a trace following the corresponding path in the CA_G pre-proof containing a progression point for each progression point in the CRTC_G^ω+A trace. From this it follows that if the CRTC_G^ω+A pre-proof satisfies the (RTC_G^ω) global trace condition, then its translation satifies the CA_G global trace condition. Finally, we derive $\Gamma^\beta \Rightarrow \Delta^\beta$ from $\Gamma^{\bar\beta} \Rightarrow \Delta^\beta$ as described above. $\qquad \square$

Theorem 8. $\vdash_{\mathsf{CRTC}_G^\omega + \mathsf{A}} \Gamma \Rightarrow \Delta$ *iff* $\vdash_{\mathsf{CA}_G} \Gamma^\beta \Rightarrow \Delta^\beta$.

Proof. By Corollary 3 and Lemma 8. $\qquad \square$

These results allow us to show an equivalence between the finitary and cyclic systems for TC with arithmetic.

Theorem 9. RTC_G+A *and* CRTC_G^ω+A *are equivalent.*

Proof. The fact that RTC_G+A $\subseteq \mathsf{CRTC}_G^\omega$+A follows immediately from Theorem 5. For the converse, suppose $\Gamma \Rightarrow \Delta$ is provable in CRTC_G^ω+A. By Theorem 8 we get

that $\vdash_{\mathsf{CA}_G} \Gamma^\beta \Rightarrow \Delta^\beta$. Using the equivalence between CA_G and PA_G, we obtain $\vdash_{\mathsf{PA}_G} \Gamma^\beta \Rightarrow \Delta^\beta$. Then we conclude using Theorem 7(2). □

Note that the result above can easily be extended to show that adding the same set of additional axioms to both $\mathsf{RTC}_G+\mathsf{A}$ and $\mathsf{CRTC}_G^\omega+\mathsf{A}$ results in equivalent systems. Also note that in the systems with pairs, to embed arithmetics there is no need to explicitly include addition and its axioms. Thus, by only including the signature $\{0, \mathsf{s}\}$ and the corresponding axioms for it we can obtain that $\langle\mathsf{RTC}\rangle_G+\mathsf{A}$ and $\langle\mathsf{CRTC}\rangle_G^\omega+\mathsf{A}$ are equivalent.

In [3], the equivalence result of [24] was improved to show it holds for any set of inductive predicates containing the natural number predicate N. On the one hand, our result goes beyond that of [3] as it shows the equivalence for systems with a richer notion of inductive definition, due to the expressiveness of TC. On the other hand, TC does not support restricting the set of inductive predicates, i.e. the RTC operator may operate on any formula in the language. To obtain a finer result which corresponds to that of [3] we need to further explore the transformations between proofs in the two systems. This is left for future work.

4.2.2   The General Case

As mentioned, the general equivalence conjecture between LKID and CLKID^ω was refuted in [2], by providing a concrete example of a statement which is provable in the cyclic system but not in the explicit one. The statement (called 2-Hydra) involves a predicate encoding a binary version of the 'hydra' induction scheme for natural numbers given in [19], and expresses that every pair of natural numbers is related by the predicate.[6] However, a careful examination of this counter-example reveals that it only refutes a strong form of the conjecture, according to which both systems are based on the same set of productions. In fact, already in [2] it is shown that if the explicit system is extended by another inductive predicate, namely one expressing the $\leq$ relation, then the 2-Hydra counter-example becomes provable. Therefore, the less strict formulation of the question, namely whether for any proof in $\mathsf{CLKID}^\omega_\phi$ there is a proof in $\mathsf{LKID}_{\phi'}$ for some $\phi' \supseteq \phi$, has not yet been resolved. Notice that in TC the equivalence question is of this weaker variety, since the RTC operator 'generates' all inductive definitions at once. That is, there is no *a priori* restriction on the inductive predicates one is allowed to use. Indeed, the 2-Hydra counter-example from [2] can be expressed in $\mathcal{L}_{\mathsf{RTC}}$ and proved in CRTC_G^ω. However, this does not produce a counter-example for TC since it is also provable in RTC_G, due to the fact that $s \leq t$ is definable via the RTC formula $(RTC_{w,u} \mathsf{s}\, w = u)(s, t)$.

Despite our best efforts, we have not yet managed to settle this question, which appears to be harder to resolve in the TC setting. One possible approach to solving it is the semantical one, i.e. exploiting the fact that the explicit system

---

[6] In fact, the falsifying Henkin model constructed in [2] also satisfies the '0-axiom' $(\forall x.0 \neq \mathsf{s}\, x)$, and the '$\mathsf{s}$-axiom' $(\forall x, y.\, \mathsf{s}\, x = \mathsf{s}\, y \rightarrow x = y)$ stipulating injectivity of the successor function, and so the actual counter-example to equivalence is the sequent: $(0, \mathsf{s})$-axioms $\Rightarrow$ 2-Hydra.
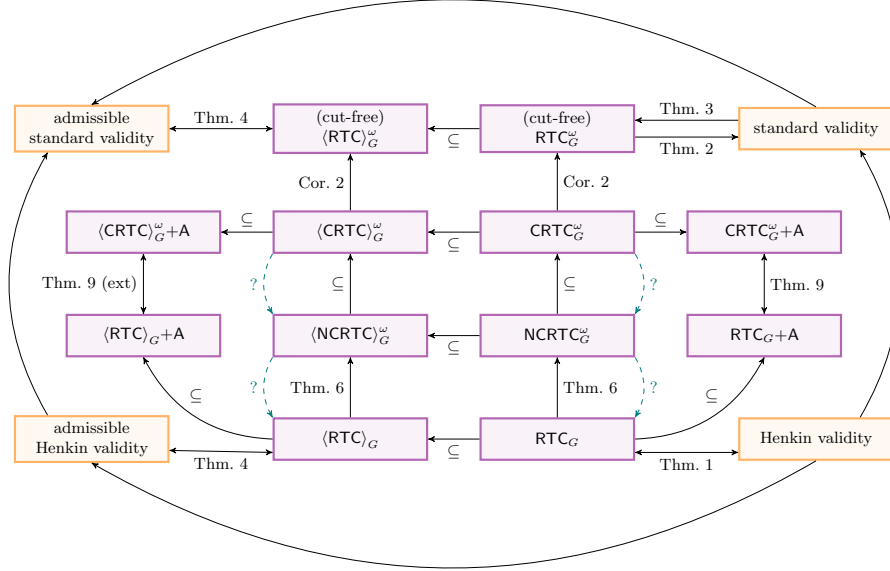
Fig. 6: Diagrammatic Summary of our Results.

is known to be sound w.r.t. Henkin semantics. This is what was done in [2]. Thus, to show strict inclusion one could construct an alternative statement that is provable in CRTC_G^ω whilst also demonstrating a Henkin model for TC that is not a model of the statement. However, constructing a TC Henkin model appears to be non-trivial, due to its rich inductive power. In particular, it is not at all clear whether the structure that underpins the LKID counter-model for 2-Hydra admits a Henkin model for TC. Alternatively, to prove equivalence, one could show that CRTC_G^ω is also sound w.r.t. Henkin semantics. Here, again, proving this does not seem to be straightforward.

In our setting, there is also the question of the inclusion of CRTC_G^ω in NCRTC_G^ω, which amounts to the question of whether overlapping cycles can be eliminated. Moreover, we can ask if NCRTC_G^ω is included in RTC_G, independently of whether this also holds for CRTC_G^ω. Again, the semantic approach described above may prove fruitful in answering these questions.

5 Conclusions and Future Work

We developed a natural infinitary proof system for transitive closure logic which is cut-free complete for the standard semantics and subsumes the explicit system. We further explored its restriction to cyclic proofs which provides the basis for an effective system for automating inductive reasoning. In particular, we syntactically identified a subset of cyclic proofs that is Henkin-complete. A summary of the proof systems we have studied in this paper, and their interrelationships, is shown in Figure 6. Where an edge between systems is labelled with an inclusion $\subseteq$, this

signifies that a proof in the source system is already a proof in the destination system.

As mentioned in the introduction, as well as throughout the paper, this research was motivated by other work on systems of inductive definitions, particularly the LKID framework of [6], its infinitary counterpart LKID^ω, and its cyclic subsystem CLKID^ω. In terms of the expressive power of the underlying logic, TC (assuming pairs) subsumes the inductive machinery underlying LKID. This is because for any inductive predicate P of LKID, there is an $\mathcal{L}_{\mathsf{RTC}}$ formula ψ such that for every standard admissible structure M for $\mathcal{L}_{\mathsf{RTC}}$, P has the same interpretation as ψ under M. This is due to Thm. 3 in [1] and the fact that the interpretation of P must necessarily be a recursively enumerable set. As for the converse inclusion, for any positive $\mathcal{L}_{\mathsf{RTC}}$ formula there is a production of a corresponding LKID inductive definition. However, the RTC operator can also be applied on complex formulas (whereas LKID productions only consider atomic predicates). This indicates that TC might be more expressive. It was noted in [6, p. 1180] that complex formulas may be handled by stratifying the theory of LKID, similar to [20], but the issue of relative expressiveness of the resulting theory is not addressed. While we strongly believe it is the case that TC is strictly more expressive than the logic of LKID, proving so is left for future work. Also left for future research is establishing the comparative status of the corresponding formal proof systems.

In addition to the open question of the (in)equivalence of RTC_G and CRTC_G^ω in the general case, discussed in Section 4.2, several other questions and directions for further study naturally arise from the work of this paper. An obvious one would be to implement our cyclic proof system in order to investigate the practicalities of using TC logic to support automated inductive reasoning. More theoretically it is already clear that TC logic, as a framework, diverges from existing systems for inductive reasoning (e.g. LKID) in interesting, non-trivial ways. The uniformity provided by the transitive closure operator may offer a way to better study the relationship between implicit and explicit induction, e.g. in the form of cuts required in each system, or the relative complexity of proofs that each system admits. Moreover, it seems likely that *co*inductive reasoning can also be incorporated into the formal system. Determining whether, and to what extent, these are indeed the case is left for future work.

References

1. Avron, A.: Transitive Closure and the Mechanization of Mathematics. In: Kamareddine, F.D. (ed.) Thirty Five Years of Automating Mathematics, Applied Logic Series, vol. 28, pp. 149–171. Springer, Netherlands (2003), doi:`10.1007/978-94-017-0253-9_7`
2. Berardi, S., Tatsuta, M.: Classical System of Martin-Löf's Inductive Definitions Is Not Equivalent to Cyclic Proof System. In: Proceedings of the 20th International Conference on Foundations of Software Science and Computation Structures, FOSSACS 2017, Uppsala, Sweden, April 22–29, 2017. pp. 301–317. Springer Berlin Heidelberg, Berlin, Heidelberg (2017), doi:`10.1007/978-3-662-54458-7_18`

3. Berardi, S., Tatsuta, M.: Equivalence of Inductive Definitions and Cyclic Proofs Under Arithmetic. In: Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20–23, 2017. pp. 1–12 (2017), doi:`10.1109/LICS.2017.8005114`

4. Brotherston, J.: Formalised Inductive Reasoning in the Logic of Bunched Implications. In: Proceedings of Static Analysis, 14th International Symposium, SAS 2007, Kongens Lyngby, Denmark, August 22–24, 2007. pp. 87–103 (2007), doi:`10.1007/978-3-540-74061-2_6`

5. Brotherston, J., Bornat, R., Calcagno, C.: Cyclic Proofs of Program Termination in Separation Logic. In: Proceedings of the 35th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2008, San Francisco, California, USA, January 7–12, 2008. pp. 101–112 (2008), doi:`10.1145/1328438.1328453`

6. Brotherston, J., Simpson, A.: Sequent Calculi for Induction and Infinite Descent. Journal of Logic and Computation 21(6), 1177–1216 (2010), doi:`10.1093/logcom/exq052`

7. Buss, S.R.: Handbook of Proof Theory. Studies in Logic and the Foundations of Mathematics, Elsevier Science (1998)

8. Cohen, L.: Ancestral Logic and Equivalent Systems. Master's thesis, Tel-Aviv University, Israel (2010)

9. Cohen, L.: Completeness for Ancestral Logic via a Computationally-Meaningful Semantics. In: Proceedings of the 26th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX 2017, Brasília, Brazil, September 25–28, 2017. pp. 247–260 (2017), doi:`10.1007/978-3-319-66902-1_15`

10. Cohen, L., Avron, A.: Ancestral Logic: A Proof Theoretical Study. In: et al., U.K. (ed.) Logic, Language, Information, and Computation, Lecture Notes in Computer Science, vol. 8652, pp. 137–151. Springer (2014), doi:`10.1007/978-3-662-44145-9_10`

11. Cohen, L., Avron, A.: The Middle Ground–Ancestral Logic. Synthese pp. 1–23 (2015), doi:`10.1007/s11229-015-0784-3`

12. Cohen, L., Rowe, R.N.S.: Uniform Inductive Reasoning in Transitive Closure Logic via Infinite Descent. In: Proceedings of the 27th EACSL Annual Conference on Computer Science Logic, CSL 2018, September 4–7, 2018, Birmingham, UK. pp. 16:1–16:17 (2018), doi:`10.4230/LIPIcs.CSL.2018.16`

13. Cook, S.A., Reckhow, R.A.: The Relative Efficiency of Propositional Proof Systems. The Journal of Symbolic Logic 44(1), 36–50 (1979)

14. Courcelle, B.: Fundamental Properties of Infinite Trees. Theor. Comput. Sci. 25, 95–169 (1983), doi:`10.1016/0304-3975(83)90059-2`

15. Das, A., Pous, D.: A Cut-Free Cyclic Proof System for Kleene Algebra. In: Proceedings of the 26th International Conference Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX 2017, Brasília, Brazil, September 25–28, 2017. pp. 261–277 (2017), doi:`10.1007/978-3-319-66902-1_16`

16. Gentzen, G.: Untersuchungen über das Logische Schließen. I. Mathematische Zeitschrift 39(1), 176–210 (1935), doi:`10.1007/BF01201353`

17. Henkin, L.: Completeness in the Theory of Types. Journal of Symbolic Logic 15(2), 81–91 (1950), `http://www.jstor.org/stable/2266967`

18. Kashima, R., Okamoto, K.: General Models and Completeness of First-order Modal μ-calculus. Journal of Logic and Computation 18(4), 497–507 (2008), doi:`10.1093/logcom/exm077`

19. Kirby, L., Paris, J.: Accessible Independence Results for Peano Arithmetic. Bulletin of the London Mathematical Society 14(4), 285–293 (1982), doi:`10.1112/blms/14.4.285`

20. Martin-Löf, P.: Hauptsatz for the Intuitionistic Theory of Iterated Inductive Definitions. In: Fenstad, J.E. (ed.) Proceedings of the Second Scandinavian Logic Symposium, Studies in Logic and the Foundations of Mathematics, vol. 63, pp. 179–216. Elsevier (1971), doi:`10.1016/S0049-237X(08)70847-4`

21. Martin-Löf, P., Sambin, G.: Intuitionistic Type Theory, vol. 9. Bibliopolis Napoli (1984)

22. Rowe, R.N.S., Brotherston, J.: Automatic Cyclic Termination Proofs for Recursive Procedures in Separation Logic. In: Proceedings of the 6th ACM SIGPLAN Conference on Certified Programs and Proofs, CPP 2017, Paris, France, January 16–17, 2017. pp. 53–65 (2017), doi:`10.1145/3018610.3018623`

23. Santocanale, L.: A Calculus of Circular Proofs and Its Categorical Semantics. In: Nielsen, M., Engberg, U. (eds.) Proceedings of the 5th International Conference on Foundations of Software Science and Computation Structures, FOSSACS 2002, Grenoble, France, April 8–12, 2002. pp. 357–371. Springer Berlin Heidelberg, Berlin, Heidelberg (2002), doi:`10.1007/3-540-45931-6_25`

24. Simpson, A.: Cyclic Arithmetic Is Equivalent to Peano Arithmetic. In: Proceedings of the 20th International Conference on Foundations of Software Science and Computation Structures, FOSSACS 2017, Uppsala, Sweden, April 22–29, 2017. pp. 283–300 (2017), doi:`10.1007/978-3-662-54458-7_17`

25. Sprenger, C., Dam, M.: On the Structure of Inductive Reasoning: Circular and Tree-Shaped Proofs in the μCalculus. In: Gordon, A.D. (ed.) Proceedings of the 6th International Conference on Foundations of Software Science and Computation Structures, FOSSACS 2003, Warsaw, Poland, April 7–11, 2003. pp. 425–440. Springer Berlin Heidelberg, Berlin, Heidelberg (2003), doi:`10.1007/3-540-36576-1_27`

26. Takeuti, G.: Proof Theory. Courier Dover Publications (1987)

27. Tellez, G., Brotherston, J.: Automatically Verifying Temporal Properties of Pointer Programs with Cyclic Proof. In: Proceedings of the 26th International Conference on Automated Deduction, CADE 26, Gothenburg, Sweden, August 6–11, 2017. pp. 491–508 (2017), doi:`10.1007/978-3-319-63046-5_30`