Tel Aviv University

School of Mathematical Sciences

The Raymond and Beverly Sackler Faculty of Exact Sciences

Ancestral Logic and Equivalent Systems

This thesis is submitted in partial fulfillment of the requirements towards the M.Sc. degree

by

Liron Cohen

This thesis was prepared under the supervision of

Prof. Arnon Avron

December, 2010

Acknowledgments

First and foremost, I would like to express my deep gratitude, respect, and thanks to my advisor Prof. Arnon Avron. Every step of this thesis was followed by his unconditional guidance and constant encouragement. I thank him for his undaunted professionalism and his uncompromising excellence.

A special thank you to my parents, Gavriel and Ester.

I would also like to thank Barak for his everlasting loving support.

Abstract

Many efforts have been made in recent years to construct computerized systems for mechanizing general mathematical reasoning. Most of the systems are based on logics which are stronger than first-order logic (FOL). However, there are good reasons to avoid using full second-order logic (SOL) for this task. In this work we investigate a logic which is intermediate between FOL and SOL, and seems to be a particularly attractive alternative to both: ancestral logic. This is the logic which is obtained from FOL by augmenting it with the transitive closure operator.

Our investigation of this logic focuses mainly on two crucial aspects of using a formal logical system: its expressive power and corresponding proof systems. The expressive power of ancestral logic is determined by comparing it with that of several other logics which are intermediate between FOL and SOL. The proof-theoretical study is done by presenting, investigating, and comparing two natural Gentzen-type proof systems which are sound for ancestral logic.

Contents

1 Introduction

Many efforts have been made in recent years to construct systems for formalizing mathematical reasoning (see [22, 23]). Most of these systems go beyond first-order logic (FOL), because the latter is too week for this task. Thus, in FOL one cannot even give a categorical characterization of the most basic concept of mathematics - the natural numbers. Hence it does not seem to be a suitable framework for formalizing mathematics.

While FOL is too weak, the choice to use second-order logic (SOL) for this task has many disadvantages. First, SOL has doubtful semantics, in the sense that it is based on debatable ontological commitments. Moreover, it does not seem satisfactory that dealing with basic notions (such as the natural numbers) requires using the strong notions involved in SOL, such as quantifying over all subsets of infinite sets. Second, SOL is difficult to deal with from a proof-theoretical point of view.

The above considerations lead to the conclusion that the most suitable framework for mechanizing mathematical reasoning should be provided by some logic which is intermediate between FOL and SOL. There are several natural candidates for this task that have been suggested in the literature, such as weak second-order logic, ω-logic, etc. We believe that the best one is ancestral logic - the logic obtained by adding to first-order logic the concept of transitive closure of a given relation. The expressive power of this logic is equivalent to that of some of the other candidates (see chapter 4), yet there are several reasons to prefer it over the others. One of them is that it seems like the easiest choice from a *proof-theoretic* point of view. Another important reason is simply the simplicity of the notion of transitive closure. Any person, even with no mathematical background whatsoever, can easily grasp the concept of the ancestor of a given person (or in other words, the idea of transitive closure of a certain binary relation). Here are some examples of the use of transitive closure in every day life:

1. The transitive closure of the relation "x is a child of y" is: "x is a descendant of y" . We often use this transitive relation to make inferences, such as: if a disease is hereditary, i.e. transferred from parent to child, and one of my ancestors had this disease, then I'll have this disease too.

2. The transitive closure of the relation "there is a regular direct flight from airport x to airport y" is: "it is possible to reach y from x by one

or more regular flights". More generally, the concept of connectivity in graph theory is a direct application of the notion of transitive closure.

3. Another mathematical example: Understanding the concept of the natural numbers is basically understanding that every number is a descendant of zero throw the successor relation. Also understanding the concept of a wwf in formal logic involves applying certain operations again and again starting from a class of atoms. Thus, the understanding of basic arithmetic and basic logic relies on the understanding of the idea of the transitive closure.

The examples above (especially the last one) show that any system designed for capturing the ability to do mathematics must provide the means to create the transitive closure of a relation and to make appropriate inferences regarding it. The examples also show that our basic understanding of the transitive closure operator involves two components: that we can construct a new binary relation from a given one (the transitive closure of the given relation), and that if a certain property is hereditary between objects in a given relation, then it will also be hereditary between objects which are related by the new relation. In this work we present related proof systems for ancestral logic which are based on these observations. We then provide some evidence that these systems are a convenient framework for formalizing mathematics. Also, we will attempt to show that these systems are adequate for the use of ancestral logic.

At this point, it is important to note that a great amount of work has been devoted in the past to ancestral logic in finite model theory and related areas of computer science. Unfortunately, that work is mostly irrelevant for the task of formalizing mathematics, since it mostly deals with finite structures. Moreover, most of that research has been done from a purely model-theoretic point of view, while here we seek to find useful *proof systems* for ancestral logic.

Thesis organization

The rest of this thesis consists of 5 chapters:

- Chapter 2 provides some definitions and notations which will be necessary throughout the thesis.
- In Chapter 3 the formal definitions of the transitive closure operator and ancestral logic are given. Then, some of the most important model-theoretic properties of ancestral logic are presented and its expressive power is described.
- Chapter 4 describes some alternative logics between first-order logic and second-order logic. After introducing the logics, we show that they indeed all offer stronger expressive power than first-order logic, but weaker than second-order logic. Next, we compare the expressive power of these logics using three different types of comparison. From this comparison we conclude that all of them are essentially equivalent to ancestral logic.
- Chapter 5 contains the main new results of this work. It deals with ancestral logic from a proof-theoretic point of view. We start by reviewing previous Hilbert-style proof systems which have been suggested in the literature. Then, two related natural Gentzen-style systems which are sound for ancestral logic is presented: one for the reflexive transitive closure and one for the non-reflexive one. This is followed by exploring some properties of these systems which suggest that the system for the reflexive transitive closure is superior to that of the non-reflexive one from a proof theoretical point of view. We end by discussing basic proof-theoretical questions concerning the systems such as: cut-elimination and constructive consistency proof.
- Chapter 6 presents some directions for further research.

2 Preliminaries

In what follows σ is a first-order signature with equality, $L^1(\sigma)$ is the first-order language based on σ and φ, ψ are meta-variables for formulas.

To avoid confusion regarding brackets, we use "(", ")" for brackets in a formal language, and "[", "]" for brackets in the meta-language.

Notation 2.1. $Fv[\varphi]$ denotes the set of all free variables occurring in φ.

Notation 2.2. Let φ be a formula, $t_1, ..., t_n$ terms, and $x_1, ...x_n$ distinct variables in some language L. Then $\varphi\left\{\frac{t_1}{x_1}, ..., \frac{t_n}{x_n}\right\}$ denotes the formula obtained from φ by simultaneously substituting t_i for each free occurrence of x_i in φ. We assume that $t_1, ..., t_n$ are free for $x_1, ...x_n$ in φ.

Notation 2.3. Let φ be a formula in prenex normal form. Then $\hat{\varphi}$ denotes the matrix of φ.

Notation 2.4. $\sigma_{PA} = \{0, s, +, *\}$ where 0 is a constant symbol, s is a unary function symbol, and $+, *$ are binary function symbols.

Definition 2.5. The first-order axiom system PA (Peano Arithmetic) consists of the axioms below. The first six are axioms for the successor, addition and multiplication, respectively, and the last one is a scheme for induction.

$$\forall x\,(s\,(x) \neq 0) \tag{2.1}$$

$$\forall x \forall y\,(s\,(x) = s\,(y) \rightarrow x = y) \tag{2.2}$$

$$\forall x\,(x + 0 = x) \tag{2.3}$$

$$\forall x \forall y\,(x + s\,(y) = s\,(x + y)) \tag{2.4}$$

$$\forall x\,(x * 0 = 0) \tag{2.5}$$

$$\forall x \forall y\,(x * s\,(y) = x * y + x) \tag{2.6}$$

$$\varphi\left\{\tfrac{0}{x}\right\} \wedge \forall x\left(\varphi \rightarrow \varphi\left\{\tfrac{s(x)}{x}\right\}\right) \rightarrow \forall x \varphi \tag{2.7}$$

Definition 2.6. A *structure* for $L^1(\sigma)$ is an ordered pair $M =< D, I >$, where D is a non-empty set of elements, called the domain of the structure, and I is an interpretation function on σ such that:

- For a constant symbol c in σ: $I[c] \in D$.

- For an n-ary function symbol f in σ: $I\left[f\right] \in D^n \to D$.
- For an n-ary predicate symbol P in σ: $I\left[P\right] \subseteq D^n$.

Notation 2.7. $\mathcal{N} =< \mathbb{N}, I >$is the standard structure for σ_{PA}, i.e. $I\left[0\right]$ is zero, $I\left[s\right]$ is the successor function, $I\left[+\right]$ is the addition function and $I\left[*\right]$ is the multiplication function.

Definition 2.8. Let $M =< D, I >$ be a structure for $L^1\left(\sigma\right)$. An *assignment* in M is a function v from the set of variables to D. v is then extended to be a function from the set of all terms in $L^1\left(\sigma\right)$ to D in the following way:

- For c a constant symbol in σ: $v\left[c\right] = I\left[c\right]$.
- For f an n-ary function symbol in σ and $t_1, ..., t_n$ terms: $v\left[f\left(t_1, ..., t_n\right)\right] = I\left[f\right]\left[v\left[t_1\right], ..., v\left[t_n\right]\right]$.

Definition 2.9. An *x-variant of an assignment* v is an assignment which differs from v at most in the value it assigns to x. We denote by $v\left[x := a\right]$ the x-variant of v which assigns to x the element a from D.

Definition 2.10. Let σ and σ' be signatures such that $\sigma' \supseteq \sigma$. A structure $M' =< D', I' >$ for $L^1\left(\sigma'\right)$ is said to be an *expansion* of the structure $M =< D, I >$ for $L^1\left(\sigma\right)$ iff $D = D'$ and for each symbol k in σ, $I[k] = I'[k]$.

In this paper we consider a logic $\mathcal{L}$ as a couple consisting of two functions on signatures: $\mathcal{L} =< L, \vDash_{\mathcal{L}}>$. L assigns to each first-order signature a formal language, and $\vDash_{\mathcal{L}}$ assigns to each first-order signature a collection of pairs of the form $\left\langle \left\langle M, v\right\rangle, T\right\rangle$ where M is a structure for σ, v is an assignment in M, and T is a set of formulas in the language that L assigns to σ. Note that this collection must satisfy certain properties that we are not going to specify here. Formally, we write $\mathcal{L}^\sigma =< L\left(\sigma\right), \vDash_{\mathcal{L}}\left(\sigma\right) >$ to denote the logic where $L\left(\sigma\right)$ is the language which the function L assigns to σ, and $\vDash_{\mathcal{L}}\left(\sigma\right)$ is the mentioned collection of pairs assigned to σ (the semantic satisfaction relation). When there will be no danger of confusion we will omit σ from the second element of the pair and write only $\vDash_{\mathcal{L}}$. Instead of $\left\langle \left\langle M, v\right\rangle, T\right\rangle \in\vDash_{\mathcal{L}}$ we write $M, v \vDash_{\mathcal{L}} T$. The consequence relation induced by the satisfaction relation is: $T \vdash_{\mathcal{L}} \varphi$ iff for each couple $\left\langle M, v\right\rangle$, if $\left\langle \left\langle M, v\right\rangle, T\right\rangle \in\vDash_{\mathcal{L}}$, then $\left\langle \left\langle M, v\right\rangle, \left\{\varphi\right\}\right\rangle \in\vDash_{\mathcal{L}}$.

Definition 2.11. A logic $\mathcal{L}_2$ is said to be an *extension* of a logic $\mathcal{L}_1$ iff for each first-order signature σ, $L_1(\sigma) \subseteq L_2(\sigma)$, and for each structure M for σ, an assignment v in M, and a set T of formulas in $L_1(\sigma)$: $M, v \vDash_{\mathcal{L}_2} T$ if $M, v \vDash_{\mathcal{L}_1} T$.

Definition 2.12. *Second-order logic* is the following extension of first-order logic. For each σ, the language $L_{SO}(\sigma)$ is obtained by augmenting $L^1(\sigma)$ by new relations variables: for each $i \in \mathbb{N}$, $R_0^i, R_1^i, ...$ are relations variables symbols of arity i, and functions variables: for each $i \in \mathbb{N}$, $F_0^i, F_1^i, ...$ are functions variables symbols of arity i. To the usual definition of wffs of $L^1(\sigma)$ we add the following clauses:

- If F^i is a i-ary function variable and $t_1, ..., t_k$ are terms, then the expression $F^i t_1, ..., t_k$ is a term.
- If R^i is a i-ary relation variable and $t_1, ..., t_k$ are terms, then the expression $R^i t_1, ..., t_k$, also denoted by $(t_1, ..., t_k) \in R^i$, is an atomic formula.
- If X is a relation variable or a function variable, and φ is a formula, then $\forall X \varphi$ and $\exists X \varphi$ are formulas.

The *standard semantics* for second-order logic is defined as follows: the interpretations of the first-order quantifiers and the logical connectives are the same as in first-order logic. An assignment assigns to each relation variable R^i a i-ary relation on the domain, and to each function variable F^i a i-ary function on the domain. The satisfaction relation is then extended in the natural way.

For simplicity, we usually treat second-order logic without the addition of the functions variables, retaining only the relations variables. This is not an essential restriction since n-ary functions can be considered as special $(n+1)$-ary relations.

3 First-order languages augmented by the TC operator

We present new types of logics created by adding to standard first-order logic the transitive closure operator. Before doing this, we first should have a basic understanding of what this operator means. Thus, we start by formally defining the transitive closure of a binary relation.

Definition 3.1. Let X be a set and $R \subseteq X \times X$ be a binary relation on X. The *transitive closure operator* TC_R of the relation R is the smallest relation $TC_R \subseteq X \times X$ such that the following holds:

1. $R \subseteq TC_R$

2. TC_R is transitive.

The relation TC_R exists for any binary relation R. To see this, note that there exists at least one transitive relation containing R, the trivial one: $X \times X$. Furthermore, the intersection of any family of transitive relations is again transitive. Hence, the transitive closure of R is the intersection of all transitive relations containing R.

Definition 3.1 is an impredicative definition. A more constructive, predicative characterization can be obtain as follows:

Proposition 3.2. *Let X be a set and $R \subseteq X \times X$ be a binary relation on X. The* transitive closure operator TC_R *of the relation R is defined by*

$$TC_R = \bigcup_{n \in \mathbb{N}^+} R^n$$

where R^n is defined by

$$R^n = \begin{cases} R & if\ n = 1 \\ R^{n-1} \circ R & otherwise \end{cases}$$

Next, we embed this concept into a logical framework. The essential idea is that one may treat a first-order formula with two (assigned) free variables as a definition of a binary relation. Below are the corresponding formal definitions of a first-order logic augmented by the TC operator, and its semantics.

Definition 3.3.

- Let σ be a signature for a first-order language with equality. *The language* $L_{TC}(\sigma)$ is defined as the first-order language based on σ, $L(\sigma)$, with the addition of the TC operator defined by:
  for any formula φ in $L(\sigma)$, x, y distinct variables, and s, t terms, $(TC_{x,y}\varphi)(s,t)$ is a formula in $L_{TC}(\sigma)$. The free occurrences of x and y in φ become bound in this formula.

- Let $M =< D, I >$ be a structure for σ and v an assignment in M. The pair $\langle M, v\rangle$ is said to *satisfy* $(TC_{x,y}\varphi)(s,t)$ (denoted by $M, v \models_{\mathcal{L}_{TC}} (TC_{x,y}\varphi)(s,t)$) if and only if there exists $a_o, ..., a_n \in D$ such that $v[s] = a_0$, $v[t] = a_n$, and φ is satisfied by M and $v[x := a_i, y := a_{i+1}]$ for $0 \leq i \leq n-1$. The logic obtained is called *ancestral logic* and it is denoted by $\mathcal{L}_{TC}$.

The first (as far as we know) to suggest expanding first-order logic by the TC operator was R.M. Martin in [4, 5]. Actually, Martin used a generalized form of the transitive closure operator. He expended first-order logic by adding for each $n \in \mathbb{N}$ a TC^n operator which when applied to an $2n$-ary predicate produce a new $2n$-ary predicate.

Definition 3.4.

- Let σ be a signature for a first-order language with equality. *The language* $L_{TC^*}(\sigma)$ is defined as the first-order language based on σ, $L(\sigma)$, with the addition of the TC^n operators for each $n \in \mathbb{N}$, where the latter is defined as follows:
  for any formula φ in $L(\sigma)$, $x_1, ..., x_n, y_1, ..., y_n$ distinct variables, and $s_1, ..., s_n, t_1, ..., t_n$ terms, $\left(TC^n_{x_1,...,x_n,y_1,...,y_n}\varphi\right)(s_1, ..., s_n, t_1, ..., t_n)$ is a formula in $L_{TC_n}(\sigma)$.[1]The *language* $L_{TC_n}(\sigma)$ is the language obtained by adding only the TC^n operator for a specific n.

- The semantics for this new types of formulas is a trivial generalization of Definition 3.3, only this time we need to refer to vectors of length n of terms or variables. We shall denote the logic obtained by adding to first-order logic all the operators TC^n for $n \in \mathbb{N}$ by $\mathcal{L}_{TC^*}$, and the logic obtained by adding only the TC^n operator for a specific n by $\mathcal{L}_{TC_n}$.

[1] Note that the TC operator of definition 3.3 is TC^1.

In [3], J. Myhill presented a first-order logic augmented only by the operator TC^1, but together with the introduction of ordered pairs into the language. The expressive power of the logic presented by Martin turns out to be the same as that of the logic presented by Myhill.

In the semantics $(TC_{x,y}\varphi)(s,t)$ requires that there should be at least one φ-step between s and t. However, both Martin and Myhill chose to take as primitive the *reflexive* TC operator, RTC.

Definition 3.5. For each first-order signature σ, the language $L_{RTC}(\sigma)$ is defined in the same way as $L_{TC}(\sigma)$ replacing TC by RTC. Let $M =< D, I >$ be a structure for σ and v an assignment in M. $\langle M, v\rangle$ is said to *satisfy* $(RTC_{x,y}\varphi)(s,t)$ iff $s = t$ or there exists $a_o, ..., a_n \in D$ such that $v[s] = a_0$, $v[t] = a_n$, and φ is satisfied by M and $v[x := a_i, y := a_{i+1}]$ for $0 \leq i \leq n-1$. Similarly, we denote the obtained logic by $\mathcal{L}_{RTC}$.

The definitions of $\mathcal{L}_{RTC^*}$ and $\mathcal{L}_{RTC_n}$ correspond to the definitions of $\mathcal{L}_{TC^*}$ and $\mathcal{L}_{TC_n}$ (see 3.4), replacing the operator TC with RTC.

Proposition 3.6. *The two forms of the transitive closure operator are definable in terms of each other.*

Proof. The reflexive transitive closure operator, RTC, is definable using the non-reflexive form by:

$$(RTC_{x,y}\varphi)(s,t) := (TC_{x,y}\varphi)(s,t) \lor s = t$$

and the non-reflexive TC operator is definable using either of the following 3 forms (which can be easily shown to be equivalent):

$$\begin{aligned}(TC_{x,y}\varphi)(s,t) :&= \exists z\left(\varphi\left\{\frac{s}{x},\frac{z}{y}\right\} \land (RTC_{x,y}\varphi)(z,t)\right)\\ &= \exists z\left((RTC_{x,y}\varphi)(s,z) \land \varphi\left\{\frac{z}{x},\frac{t}{y}\right\}\right)\\ &= \exists z\exists u\left((RTC_{x,y}\varphi)(s,z) \land \varphi\left\{\frac{z}{x},\frac{u}{y}\right\} \land (RTC_{x,y}\varphi)(u,t)\right)\end{aligned}$$

Where u, v are fresh variables.

The same connection holds also between the two generalized forms of the transitive closure: TC^n and RTC^n. □

One difference between the two forms is their ability to define quantifiers.

Proposition 3.7. *The existential quantifier can be defined using the TC operator, while it cannot be defined using the RTC operator.*

Proof. By using the TC operator we can define:

$$\exists x\varphi := \left(TC_{u,v}\left(\varphi\left\{\frac{u}{x}\right\} \vee \varphi\left\{\frac{v}{x}\right\}\right)\right)(s,t)$$

Where u, v are two fresh variables (not occurring in φ). Note that s and t can be any two terms, yet in order to keep the usual condition of not having x occur free in $\exists x\varphi$ we should take s and t to be closed terms. (This is possible of course iff the language contains a constant symbol).

To see that the existential quantifier cannot be defined using the RTC operator, take σ to consist of a constant symbol 0 and a unary predicate symbol P. We shall prove that each quantifier-free sentence ψ in $\mathcal{L}^{\sigma}_{RTC}$ is logically equivalent to one of the following four forms: $P(0)$, $\neg P(0)$, $0 = 0$ or $0 \neq 0$. The proof is carried out by induction on ψ. If ψ is an atomic sentence then it is $P(0)$ or $0 = 0$, and the claim trivially holds. Assume that for the sentences φ and ϕ the claim holds. If $\psi = \neg\varphi$ then if φ is equivalent to $P(0)$ then ψ is equivalent to $\neg P(0)$ and vice verse; and if φ is equivalent to $0 = 0$ then ψ is equivalent to $0 \neq 0$ and vice verse. If $\psi = \varphi \wedge \phi$ then there are several options. If one of φ, ϕ is equivalent to $0 \neq 0$, then ψ is equivalent to $0 \neq 0$ as well. If none of φ, ϕ are equivalent to $0 \neq 0$ and one of them, say φ, is equivalent to $0 = 0$, then ψ is equivalent to ϕ which by the induction hypothesis is equivalent to one of the four forms. If none of φ, ϕ are equivalent to $0 \neq 0$ or $0 = 0$ there can be two cases: if both are equivalent to the same sentence, then ψ is equivalent to that very sentence; otherwise, ψ is equivalent to $P(0) \wedge \neg P(0)$, i.e. to $0 \neq 0$. Similar arguments apply for the remaining connectives. If $\psi = (RTC_{x,y}\varphi)(0,0)$ for some formula φ, then by the definition of the reflexive transitive closure operator ψ is equivalent to $0 = 0$. Since $\exists x P(x)$ is obviously not logically equivalent to any of the four sentences, we get that the existential quantifier cannot be defined in the quantifier-free fragment of $\mathcal{L}^{\sigma}_{RTC}$. □

In general, the TC operator is not first-order definable. However, there are cases in which there is a first-order sentence equivalent to $(TC_{x,y}\varphi)(s,t)$. The obvious case is when φ is a valid formula, since then $(TC_{x,y}\varphi)(s,t)$ is also a valid formula. This case is a special case of the following type of formulas.

Definition 3.8. Let φ be a formula in $L^1(\sigma)$. φ is called a *transitive formula* if for every structure M for σ, every assignment v for M, and every $a, b, c \in D$: if $M, v[x := a, y := b] \models \varphi$ and $M, v[x := b, y := c] \models \varphi$, then
$M, v[x := a, y := c] \models \varphi$.

Example. The formula $P(x) \wedge P(y)$ is transitive, while $P(x) \vee P(y)$ is not.

The following theorem is obvious.

Theorem 3.9. *Let φ be a transitive formula and s,t terms in $L^1(\sigma)$. For every structure M for σ and assignment v for M :*

$$M, v \models (TC_{x,y}\varphi)(s, t) \iff M, v \models \varphi\left\{\frac{s}{x}, \frac{t}{y}\right\}$$

Corollary 3.10. *If φ is a transitive formula in $L^1(\sigma)$, then $(TC_{x,y}\varphi)(s, t)$ is definable in $L^1(\sigma)$.*

As seen in Proposition 3.7, $\exists x P(x) := (TC_{u,v}(P(u) \vee P(v)))(s, t)$. Thus, $P(u) \vee P(v)$ is an example of a non-transitive formula whose transitive closure is first-order definable. We leave it as an open problem to characterize the set of first-order logic formulas whose transitive closure is definable in first-order logic.

Next we show that the TC operator is not first-order definable in the general case. Thus by adding it to a first-order language we truly enlarge the expressive power of the logic.

Theorem 3.11. *Let $\{R, c\} \subseteq \sigma$ where R is a binary relation symbol and c a constant symbol. There is no sentence ψ in $L^1(\sigma)$ such that for any structure M for σ: $M \models \psi$ iff $M \models_{\mathcal{L}_{TC}} (TC_{x,y}R(x, y))(c, c)$.*

Proof. Assume that there is a sentence ψ in $L^1(\sigma)$ such that for each structure M for σ: $M \models_{L^1} \psi$ iff $M \models_{\mathcal{L}_{TC}} (TC_{x,y}R(x, y))(c, c)$. Next we formalize sentences that state that for each natural number n there is no path (of length n or smaller) connecting $I[c]$ to itself in the interpretation of R. Formally, define:

$$\phi_0 := \neg R(c, c)$$

$$\vdots$$

$$\phi_n := \neg\exists x_1 ... \exists x_n \left(R(c, x_1) \wedge \bigwedge_{1 \leq i \leq n-1} R(x_i, x_{i+1}) \wedge R(x_n, c)\right)$$

Informally speaking, the set $\{\phi_n \mid 0 \leq n\}$ express that there is no R-path from c to itself. Thus, the (infinite) set $T = \{\phi_n \mid 0 \leq n\} \cup \{\psi\}$ is not satisfiable. However, we will prove that any *finite* subset of T is satisfiable. Consider an arbitrary finite subset $T' \subset T$. Then, there is a $0 \leq n_0$ such that: $T' \subseteq T^* = \{\phi_n \mid 0 \leq n \leq n_0\} \cup \{\psi\}$. The satisfiablity of T^* surely implies the satisfiablity of T'. To prove that T^* has a model we look at a structure whose domain is $\{0, 1, ..., n_0\}$ so that $I[c] = 0$, $I[R] = \{(x, y) \mid y = x + 1 \; or \, x = n_0 + 1 \, and \, y = 0\}$. We have to show that $M \models T^*$, i.e. that M is a model for each $\theta \in T^*$.

- If $\theta = \psi$ then M is a model of θ according to the definition of the structure and the TC operator.

- If $\theta = \phi_n$ for $0 \leq n \leq n_0$ then M is a model of θ iff there is no R-path of length smaller or equal to n from $I[c]$ to itself. Obviously M has this property since the R-path starting from 0 and leading back to 0 is the only R-path connecting $I[c]$ to itself, and it is of length $n_0 + 1 > n$.

To sum up, we have shown that M is a model for T^*. Since T^* is satisfiable so is T'. Using the compactness theorem for first-order logic we can conclude that T is also satisfiable, which leads to a contradiction. Thus, the transitive closure operator is not definable in first-order logic.

□

Though the TC operator cannot be defined in first-order logic, it is definable in second-order logic.

Proposition 3.12. *The RTC operator is definable in monadic second-order logic, hence also the TC operator is so definable.*

Proof. Let X be a monadic relation variable. We have that

$$\begin{gathered}(RTC_{x,y}\varphi)(s, t) \equiv \\ \forall X\left((Xs \wedge \forall x \forall y\,(\varphi(x, y) \wedge Xx \rightarrow Xy)) \rightarrow Xt\right)\end{gathered}$$

From Proposition 3.6 we get that the TC operator is also second-order definable. □

The concept of the TC operator is embedded in our understanding of the natural numbers. The fact that the reasoning needed in order to grasp the concept of the ancestral is exactly the same as needed in order to understand basic arithmetic naturally leads to exploring the expressive power of various first-order languages for arithmetic augmented by the TC operator.

Theorem 3.13. *In $\mathcal{L}_{TC}^{\{0,s\}}$, let ψ be the conjunction of the axioms for the successor function, 2.1 and 2.2. Let φ be the following axiom:*

$$\varphi := \forall x \left(x = 0 \vee \left(TC_{w,u} \left(s(w) = u\right)\right)(0, x)\right) \tag{3.1}$$

The set $T = \{\psi, \varphi\}$ categorically characterize the natural numbers.

Proof. Clearly, the structure $\mathbb{N}$ whose domain is the natural numbers and the interpretation of the constant $'0'$ is zero and and the interpretation of the function symbol s is the successor function, is a model for T. Standard arguments show that it is the only model of T (up to isomorphism). □

Although the natural numbers can be categorically characterized in $\mathcal{L}_{TC}^{\{0,s\}}$, the expressive power of this language is too weak (for example, $+$ is not definable in it, as will be shown below). We present some results regarding the expressive power of first-order languages for arithmetic with the TC operator.

Theorem 3.14.

1. *All recursive functions and relations are definable in $\mathcal{L}_{TC}^{\{0,s,+\}}$. The same result holds in $\mathcal{L}_{TC_2}^{\{0,s\}}$.*

2. *In the absence of ordered pairs one cannot define $+$ in $\mathcal{L}_{TC}^{\{0,s\}}$.*

Proof.

1. Obviously, it is sufficient to show that addition and multiplication are definable. The division relation is definable in $\mathcal{L}_{TC}^{\{0,s,+\}}$ since[2]

$$\mathbb{N} \models_{\mathcal{L}_{TC}} u \mid v \leftrightarrow$$
$$v = 0 \vee \left(TC_{x,y} \left(x + u = y\right)\right)(0, v)$$

[2] The definition in [1] has a minor inaccuracy in handling the case where $v = 0$.

We also have that

$$\mathbb{N} \models_{\mathcal{L}_{TC}} x = y^2 \leftrightarrow$$
$$y \mid (x + y) \wedge s(y) \mid (x + y) \wedge$$
$$\forall z \left((TC_{u,v} v = s(u))(z, x + y) \wedge y \mid z \to \neg (s(y) \mid z) \right)$$

This formula is obviously true in case $y \neq 0$, because $x = y^2$ iff the least common multiple of x, y is $y * (y + 1)$.On the other hand, if $y = 0$, then since 0 divides only 0, $0 \mid (x + 0)$ implies that $x = 0$. In this case $x + y = 0$, and since 0 is divided by any number (in particular by 1) we get that $s(y) \mid (x + y)$. The formula $(TC_{u,v} v = s(u))(z, x + y)$ is equivalent to $z < x + y$, which in this case is $z < 0$, and since there is no such z the last conjunct holds.

From the last observation it easily follows that $*$ is definable in $\mathcal{L}_{TC}^{\{0,s,+\}}$ as follows:

$$\mathbb{N} \models_{\mathcal{L}_{TC}} x = y * z \leftrightarrow$$
$$\exists u \exists v \exists w \left(u = y^2 \wedge v = z^2 \wedge w = (y + z)^2 \wedge w = (((u + v) + x) + x) \right)$$

The fact that the same result holds in $\mathcal{L}_{TC_2}^{\{0,s\}}$ follows from the fact that $+$ is definable in this language by:

$$\mathbb{N} \models_{\mathcal{L}_{TC_2}} x = y + z \leftrightarrow$$
$$(RTC_{x_1,x_2,y_1,y_2}(y_1 = s(x_1) \wedge y_2 = s(x_2)))(0, y, z, x)$$

In $\mathcal{L}_{TC_2}^{\{0,s\}}$ the same method can be applied to get a more direct definition of multiplication in terms of addition, for instance:

$$\mathbb{N} \models_{\mathcal{L}_{TC_2}} x = y * z \leftrightarrow$$
$$(RTC_{x_1,x_2,y_1,y_2}(y_1 = s(x_1) \wedge y_2 = x_2 + z))(s(0), z, y, x) \vee (y = 0 \wedge x = 0)$$

2. By proposition 3.12 we have that the logic $\mathcal{L}_{TC}^{\{0,s\}}$ is interpretable in the monadic second-order theory of the successor function (see [21]). The decidability of the monadic second-order theory of the successor function then implies the decidability of the set of formulas in $\mathcal{L}_{TC}^{\{0,s\}}$ which are valid in $\mathbb{N}$. Assuming that $+$ is definable in $\mathcal{L}_{TC}^{\{0,s\}}$, we get that

the set of formulas in $\mathcal{L}_{TC}^{\{0,s,+\}}$ which are valid in $\mathbb{N}$ is decidable. Yet, by (1) we have that all recursive functions and relations are definable in $\mathcal{L}_{TC}^{\{0,s,+\}}$, therefore the set of formulas which are valid in $\mathbb{N}$ in $\mathcal{L}_{TC}^{\{0,s,+\}}$ is not even arithmetical.

□

Note 3.15. The proof of the first part of 3.14(1) is a slight correction to that in [1]. The second part of 3.14(1) is essentially due to Quine [18] (only Quine used a variation of $\mathcal{L}_{TC}^{\{0,s\}}$ with ordered pairs). 3.14(2) is again due to [1].

4 Ancestral logic vs. other extensions of first-order logic

In the last chapter we described ancestral logic. However, there are other logics between first-order logic and second-order logic that might be used for the same purposes. The main strength of full second-order logic is its ability to characterize important mathematical structures and concepts that first-order logic cannot. Examples are provided by natural numbers and the concept of finitude. In this chapter we will present several other logics, in addition to those presented before, that presuppose the notion of finitude or natural number, each in a different way. After characterizing each of the logics, we show how there is a sense in which they are all equivalent. Then their expressive resources are assessed and compared to one another.

Most of the results of this chapter are stronger versions of results which appear in [6]. Theorem 4.23 is taken from [16].

4.1 The logics

Let us start by presenting the logics which will be examined in this chapter and then show how the concept of the natural number can be characterized in each of them.

Weak second-order logic

We have argued that full second-order logic has many disadvantages, yet we can examine a weaker version of it called (unsurprisingly) weak second-order logic. In this logic the relation variables and the second-order quantifiers range only over finite relations.

Definition 4.1 ($\mathcal{L}_{WSO}$)**.**

- *The language* is the same as that of second-order logic.
- *The semantic satisfaction relation* is similar to that of second-order logic with the exception that any assignment v for a structure M is restricted to assign only a finite subset of the domain to each relation variable. i.e. if X is an n-place relation variable, then $v[X]$ is a finite subset of D^n.

The concept of finitude can be expressed in $\mathcal{L}_{WSO}$. More precisely, for any formula φ such that $x \in Fv[\varphi]$, it is possible to assert that there is a finite number of elements x which satisfy φ. This can be accomplished by a formula of the form: $\exists X \forall x (Xx \leftrightarrow \varphi)$. It follows that $\mathcal{L}_{WSO}$ is indeed more expressive than first-order logic, since the latter can't express finitude.

ω-logic

Perhaps the logic that captures the notion of the natural numbers in the most straightforward manner is ω-logic. There are several different equivalent variations of ω-logic. We follow [11] and add to a first-order signature a new unary predicate N which will be used to distinct the natural numbers. The addition of this predicate allows us to quantify only over the natural numbers using $\forall x (N(x) \rightarrow ...)$ or $\exists x (N(x) \wedge ...)$.

Definition 4.2 ($\mathcal{L}_\omega$)**.**

- *The language:* for each σ, the language $L_\omega(\sigma)$ is obtained by augmenting $L^1(\sigma)$ with a new unary predicate symbol N as well as the constant symbol 0 and the unary function symbol s.
- *The semantic satisfaction relation* is defined as follows. Define M to be an ω-structure if the interpretation of N in M is isomorphic to the natural numbers and 0 and s are standardly interpreted. We restrict the usual first-order semantic satisfaction relation in $\mathcal{L}_\omega$ to deal only with ω-structures.

Cardinality logic

Another way to obtain the concept of finitude in a direct way is by adding a new type of quantifier, Q_0, called a “cardinality quantifier”. The new type of formula '$Q_0 x \varphi$' may be read “there are infinitely many x for which φ holds”.

Definition 4.3 ($\mathcal{L}_{Card}$)**.**

- *The language:* for each σ, the language $L_{Card}(\sigma)$ is obtained by augmenting $L^1(\sigma)$ with a new cardinality quantifier Q_0.
  To the usual definition of wffs of $L^1(\sigma)$ we add the following clause: if φ is a formula and x is a variable, then $Q_0 x \varphi$ is a formula.

- *The semantic satisfaction relation* of first-order logic is extended such that $M, v \models_{\mathcal{L}_{Card}} Q_0 x\varphi$ iff there are infinitely many distinct x-variant assignments, v', so that $M, v' \models \varphi$.

In $\mathcal{L}_{Card}$ we can also express finitude, for example by $\neg Q_0 x\varphi$. Thus we get that $\mathcal{L}_{Card}$ is also more expressive than first-order logic.

Henkin quantifiers

The next logic, suggested in [16], also extends first-order logic by adding new types of quantifiers. In this case we add quantifiers called Henkin quantifiers. The motivation behind these types of quantifiers is that in any formula in a prenex form, each existentially quantified variable depends on all of the universally quantified variables that come before it. Therefore, it makes sense to introduce independence between some of the bound variables in a string of quantifiers. We shall not explore here the logic obtained by adding all types of Henkin quantifiers, but a fragment of it obtained by adding only special type of Henkin quantifiers, called narrow Henkin quantifiers. This is due to the special connection between narrow Henkin quantifiers and the TC operator which will be discussed in the next sections.

Definition 4.4 ($\mathcal{L}_{NH}$)**.**

- *The language:* for each σ, the language $L_{NH}(\sigma)$ is a first-order language augmented by boolean variables $\alpha, \beta, ...$ which semantically are intended to range over $\{0,1\}$ (this means that we are dealing with a two-sorted language whose second sort is to be interpreted as $\{0,1\}$ in all structures). We add to $L^1(\sigma)$ the following narrow Henkin quantifiers:
$$\begin{pmatrix} \forall \bar{x} & \exists \alpha \\ \forall \bar{y} & \exists \beta \end{pmatrix}$$
where α, β are boolean variables and $\bar{x}, \bar{y}$ are tuples of individuals or boolean variables.

  $\bar{x}$ and $\bar{y}$ are called compatible if they are of the same length and have variables of the same type at corresponding positions. We write $\bar{x} = \bar{y}$ as an abbreviation for $\bigwedge_i (x_i = y_i)$ provided that $\bar{x}$ and $\bar{y}$ are compatible.

  To the usual definition of wffs of $L^1(\sigma)$ we add the following clause: if φ is a formula and Q_{NH} is a narrow Henkin quantifier, then $Q_{NH}\varphi$ is a formula.

- *The semantic satisfaction relation* is defined as follows: $M, v \models_{\mathcal{L}_{NH}} Q_{NH}\varphi\left(\bar{x}, \bar{y}, \alpha, \beta\right)$ iff $M, v \models \exists f \exists g \forall \bar{x} \forall \bar{y} \varphi\left(\bar{x}, \bar{y}, f\left(\bar{x}\right), g\left(\bar{y}\right)\right)$ in the standard second-order semantics. In the second-order form semantics of such a quantifier, the interpretation of f, g are functions whose range is $\{0, 1\}$.

4.2 Characterizing the natural numbers

The next task is to assess and compare the expressive power of these logics. It is known that first-order logic cannot categorically characterize the natural numbers. Let us first show that all of the logics of this chapter can.

Theorem 4.5. *Let $\mathcal{L}$ be one of the logics presented in this chapter and the previous one. In $\mathcal{L}^{\sigma_{PA}}$ there is a sentence φ such that any model of φ is isomorphic to the natural numbers (with the usual interpretation of functions and relations symbols)*[3]

Proof. In $\mathcal{L}_\omega$ we can obtain the desired characterizing sentence simply by $\forall x N\left(x\right)$. In order to achieve characterization of the natural numbers in the other logics define ψ to be the conjunction of 2.1, 2.2, 2.3 and 2.4. We must demand that the domain contains *only* $0, s(0), s(s(0)), ...$ Therefore, we shall present in each of them a formula θ such that $\psi \wedge \theta$ is the desired sentence.

In $\mathcal{L}_{TC}$ we have already seen (Theorem 3.13) that 3.1 asserts that everything is a successor-ancestor of 0.

In $\mathcal{L}_{NH}$ the same assertion can be formalized. To show this notice that the formula $\left(TC_{w,u}\left(s(w) = u\right)\right)(0, x)$ is intuitively equivalent to:

$$\bigvee_{0 \leq n} \exists z_0 ... \exists z_n \left(z_0 = 0 \wedge z_n = x \wedge s(z_0) = z_1 \wedge ... \wedge s(z_{n-1}) = z_n\right)$$

which is equivalent to:

$$\neg \exists f \left(f\left(0\right) = 1 \wedge f\left(x\right) = 0 \wedge \forall x \forall y \left(f\left(x\right) = 1 \wedge s(x) = y \rightarrow f\left(y\right) = 1\right)\right)$$

This second-order formula is equivalent to the following formula in $\mathcal{L}_{NH}$:

$$\neg \begin{pmatrix} \forall u & \exists \alpha \\ \forall v & \exists \beta \end{pmatrix} ((u = v \rightarrow \alpha = \beta) \wedge (u = 0 \rightarrow \alpha = 1) \wedge \\ (u = x \rightarrow \alpha = 0) \wedge (\alpha = 1 \wedge s(u) = v \rightarrow \beta = 1))$$

[3] These results appear in [16, 6].

Hence, the sentence in $\mathcal{L}_{TC}$ can be formalized in $\mathcal{L}_{NH}$ by:

$$\forall z \neg \begin{pmatrix} \forall x_1 & \exists y_1 \\ \forall x_2 & \exists y_2 \end{pmatrix} [(x_1 = x_2 \rightarrow y_1 = y_2) \wedge (x_1 = 0 \rightarrow y_1 = s(0)) \wedge$$
$$(x_1 = z \rightarrow y_1 = 0) \wedge (y_1 = s(0) \wedge s(x_1) = x_2 \rightarrow y_2 = s(0))]$$

In $\mathcal{L}_{Card}$ take θ to be the following sentence:

$$\forall y (\neg Q_0 x \exists z (x + z = y))$$

$\exists z (x + z = y)$ asserts that $x \leq y$, thus θ asserts that for every object x there are only finitely many objects smaller or equal to x. Let us show that this assertion together with ψ suffice. Assume that $M = \langle D, I \rangle$ is a model for $\psi \wedge \theta$. Define $g : \mathbb{N} \rightarrow D$ recursively by $g(0) = I[0]$ and $g(m+1) = I[s][g(m)]$. The function g preserves $+$. Suppose that g is not onto D, and choose $a \in D$ which is not in the image of g. From the successor axiom we find that it is possible to define a sequence $\langle x_i \mid i \in \mathbb{N} \rangle$ such that $x_0 = a$ and $I[s][x_{i+1}] = x_i$ for every $i \in \mathbb{N}$. This clearly contradicts θ. Hence g is onto, so it is an isomorphism from the structure $\mathcal{N}$ to M.

In $\mathcal{L}_{WSO}$ we add a sentence which states that for each x the set that contains all the objects smaller or equal to x is finite, which is:

$$\forall x \exists X \forall y (\exists z (x + z = y) \rightarrow Xy)$$

In each case we found a formula θ such that any model of $\psi \wedge \theta$ is isomorphic to the natural numbers. □

The last theorem leads to the following corollary.

Corollary 4.6. *The upward Lowenhiem-Skolem theorem fails for these logics and they are not compact. Moreover, any formal deductive system which is sound for one of these logics is incomplete.*

Proof. Obviously, the existence of a characterization for the natural numbers gives a refutation for the upward Lowenhiem-Skolem theorem as well as for the compactness theorem for these logics.

Let F be any effective deductive system that is sound for one of these logics. Let ϕ be the sentence that characterize the natural numbers in this logic (such a sentence exists in each of the logics due to Theorem 4.5). Let $T = \{\psi \mid \vdash_F \phi \rightarrow \psi\}$. Since F is effective, the set T is recursively enumerable, and since F is sound, $\mathcal{N} \models T$. It follows from Godel's incompleteness theorem that the collection of valid first-order sentences of arithmetic is not recursively enumerable. So let φ be a valid sentence of first-order arithmetic that is not in T. Then, $\phi \to \varphi$ is not provable in F, but it is a logical truth. Therefore, F is not complete. □

4.3 Comparing the expressive power of the logics

Next we compare the expressive power of the logics of this chapter. In the comparison theorems in this section we shall only present full proofs for theorems concerning ancestral logic, while only stating the others.

Given two logics, the intuitive question we seek to answer is whether one logic can express all the assertions that can be expressed in the other. The follow up question is obviously: when given two logics, what are the means necessary for each logic to express a certain assertion? In order to answer these questions we first have to clarify what exactly do we mean by "expressing an assertion".

Definition 4.7. Let S be a class of structures for a first-order signature σ, and let $\mathcal{L}$ be a logic. The class S is said to be *strongly definable* in $\mathcal{L}^\sigma$ if there is a set of sentences Γ in $\mathcal{L}^\sigma$, such that for every structure M for σ: $M \models_{\mathcal{L}} \Gamma$ if and only if $M \in S$. The class S is said to be *definable* in $\mathcal{L}^\sigma$ if there is a signature $\sigma' \supseteq \sigma$, and a set of sentences Γ in $\mathcal{L}^{\sigma'}$, such that for every structure M' for σ': $M' \models_{\mathcal{L}} \Gamma$ if and only if there is a structure $M \in S$ such that M' is an expansion of M.

Note. In [6], Shapiro uses a slightly different approach to what it means to define a set of structures. The difference from the last definition is that he took Γ to be a singleton. i.e. a class of structures is to be defined by a single sentence. Since any finite set of sentences can be regarded as a single sentence (the conjuncture of the sentences), the only difference between the definitions is the possibility to define a class of structures by an infinite set of sentences.

Following the last definition, an "assertion" is to be understood as a class of structures S for a signature σ, and "expressing an assertion" in a logic is to be understood as saying that S is definable or strongly definable in the logic. The definitions of definability and strong definability lead to different methods of comparison between logics.

4.3.1 Strong inclusion

We start with the question what are the conditions under which two logics should be called equivalent. The first possible answer is that the two logics have to be able to strongly define the same classes of structures. This leads to the first (and strongest) form of comparison between two logics: the existence of a signature-preserving translation between the logics.

Definition 4.8. Let $\mathcal{L}_1, \mathcal{L}_2$ be two logics ($\mathcal{L}_1$ is not $\mathcal{L}_\omega$). We say that $\mathcal{L}_2$ *strongly includes* $\mathcal{L}_1$ ($\mathcal{L}_2 \overset{s}{\geq} \mathcal{L}_1$) if given a first-order signature σ, every set of structures strongly definable in $\mathcal{L}_1^\sigma$ can be strongly defined in $\mathcal{L}_2^\sigma$.

Because of the special nature of $\mathcal{L}_\omega$, we must extend the above definition a bit to accommodate $\mathcal{L}_\omega$.

Definition 4.9. Let $\mathcal{L}$ be a logic (other than $\mathcal{L}_\omega$). We say that $\mathcal{L}$ *strongly includes* $\mathcal{L}_\omega$ if given a first-order signature σ, every set of structures strongly definable in $\mathcal{L}_\omega^\sigma$ can be strongly defined in $\mathcal{L}^{\sigma \cup \{N,s,0\}}$.

We say that $\mathcal{L}_\mathbf{1}$ and $\mathcal{L}_\mathbf{2}$ are *strongly equivalent* ($\mathcal{L}_1 \overset{s}{\equiv} \mathcal{L}_\mathbf{2}$) if both $\mathcal{L}_2 \overset{s}{\geq} \mathcal{L}_1$ and $\mathcal{L}_1 \overset{s}{\geq} \mathcal{L}_2$. We write $\mathcal{L}_2 \overset{s}{>} \mathcal{L}_1$ for $\mathcal{L}_2 \overset{s}{\geq} \mathcal{L}_1$ and $\mathcal{L}_1 \overset{s}{\ngeq} \mathcal{L}_2$.

Note. Obviously, the strong inclusion relation is transitive, i.e. if $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$ are three logics such that $\mathcal{L}_2 \overset{s}{\geq} \mathcal{L}_1$ *and* $\mathcal{L}_3 \overset{s}{\geq} \mathcal{L}_2$*, then* $\mathcal{L}_3 \overset{s}{\geq} \mathcal{L}_1$*.*

Proposition 4.10. *Let σ be a first-order signature. If for each sentence ψ in $L_1(\sigma)$, there is a sentence ψ' in $L_2(\sigma)$ such that for every structure M, $M \models_{\mathcal{L}_\mathbf{1}} \psi$ iff $M \models_{\mathcal{L}_\mathbf{2}} \psi'$, then* $\mathcal{L}_2$ strongly includes $\mathcal{L}_1$*.*

Proof. Let σ be a first-order signature and let K be a set of structures for σ strongly defined in $\mathcal{L}_1^\sigma$ by a set of sentences Γ. By the assumption, for each sentence ψ in $L_1(\sigma)$, there is a sentence ψ' in $L_2(\sigma)$ such that for every structure M, $M \models_{\mathcal{L}_\mathbf{1}} \psi$ iff $M \models_{\mathcal{L}_\mathbf{2}} \psi'$. It is easy to see that the set $\Gamma' = \{\psi' \mid \psi \in \Gamma\}$ defines the set K in $\mathcal{L}_2^\sigma$. □

Next we investigate which $\overset{s}{\geq}$ holds between $\mathcal{L}_{TC}, \mathcal{L}_{TC^*}$ and the other logics of this chapter.

Theorem 4.11. $\mathcal{L}_{TC^*} \overset{s}{>} \mathcal{L}_{TC}$.

Proof. Obviously $\mathcal{L}_{TC^*} \overset{s}{\geq} \mathcal{L}_{TC}$, since $L_{TC}(\sigma) \subseteq L_{TC^*}(\sigma)$ and their semantic satisfaction relation are the same. The converse does not hold. To see this, recall that using the signature $\sigma = \{0, s\}$ addition is definable in $\mathcal{L}^{\sigma}_{TC^*}$, whereas it is not definable in $\mathcal{L}^{\sigma}_{TC}$ (see 3.14(2)). □

Theorem 4.12. $\mathcal{L}_{WSO} \overset{s}{\geq} \mathcal{L}_{TC^*}$.

Proof. For each formula φ in $\mathcal{L}_{TC^*}$ define a formula φ' in $\mathcal{L}_{WSO}$ by induction on the complexity of φ.

- If φ is atomic then φ' is φ.
- $(\varphi \to \psi)'$ is $\varphi' \to \psi'$
- $(\neg\varphi)'$ is $\neg\varphi'$
- $(\forall x\varphi)'$ is $\forall x\varphi'$

Now we need to assign to each formula of the form $\left(TC^k_{\bar{x},\bar{y}}\varphi\right)(\bar{s},\bar{t})$ a formula in $\mathcal{L}_{WSO}$. Let us denote the formula which correlates to φ in $\mathcal{L}_{WSO}$ by φ'. We use the notation $R\bar{x}\bar{y}$ as an abbreviation for $Rx_1...x_ky_1...y_k$, $\forall\bar{x}$ as an abbreviation for $\forall x_1...\forall x_k$ and $\bar{x} = \bar{y}$ as an abbreviation for $\bigwedge_{i<k}(x_i = y_i)$. We can define the formula that correlates to $\left(TC^k_{\bar{x},\bar{y}}\varphi\right)(\bar{s},\bar{t})$ by stating that there is a finite $2k$-ary relation R such that:

- R represents a sub-relation of the relation represented by φ'.

$$\psi_1 := \forall\bar{x}\forall\bar{y}\left(R\bar{x}\bar{y} \to \varphi'(\bar{x},\bar{y})\right)$$

- R represents a graph of a function f on a subset of the domain.

$$\psi_2 := \forall\bar{x}\forall\bar{y}\forall\bar{z}\left(R\bar{x}\bar{y} \wedge R\bar{x}\bar{z} \to \bar{y} = \bar{z}\right)$$

- $\bar{t}$ is in the range of f.

$$\psi_3 := \exists\bar{x}R\bar{x}\bar{t}$$

- $\bar{s}$ is in the domain of f.

$$\psi_4 := \exists\bar{x}R\bar{s}\bar{x}$$

- $\bar{s}$ is not in the range of f.

$$\psi_5 := \neg\exists\bar{x}R\bar{x}\bar{s}$$

- if $\bar{y}$ is in the range of f and $\bar{y} \neq \bar{t}$ then $\bar{y}$ is in the domain of f.

$$\psi_6 := \forall \bar{x} \forall \bar{y} \left(R\bar{x}\bar{y} \wedge \bar{y} \neq \bar{t} \to \exists \bar{z} R\bar{y}\bar{z} \right)$$

Now, define $\left(\left(TC^k_{\bar{x},\bar{y}} \varphi \right) (\bar{s}, \bar{t}) \right)' = \exists R \left(\psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4 \wedge \psi_5 \wedge \psi_6 \right)$. A straightforward induction shows that for any φ of $\mathcal{L}^{\sigma}_{TC^*}$, if M is a structure for σ, v is an assignment to the first-order variables, and v' an assignment in $\mathcal{L}^{\sigma}_{WSO}$ that agrees with v on the first-order variables, then $M, v \models_{\mathcal{L}_{TC^*}} \varphi$ iff $M, v' \models_{\mathcal{L}_{WSO}} \varphi'$. Thus, by Proposition 4.10 we get that $\mathcal{L}_{WSO} \overset{s}{\geq} \mathcal{L}_{TC^*}$.

□

Theorem 4.13. $\mathcal{L}_{TC} \overset{s}{\geq} \mathcal{L}_{\omega}$.

Proof. Define:

$$\begin{aligned} \phi : &= \forall x \left(N(x) \to s(x) \neq 0 \right) \wedge \forall x \forall y \left(N(x) \wedge N(y) \wedge s(x) = s(y) \to x = y \right) \\ & \wedge \forall x \left(N(x) \wedge x \neq 0 \to \exists y \left(N(y) \wedge s(y) = x \right) \right) \end{aligned}$$

Conjoin ϕ with the assertion that every element in N is a descendant of 0 under the successor relation:

$$\varphi := \forall y \left(N(y) \leftrightarrow \left(TC_{w,u} \left(s(w) = u \right) \right) (0, y) \right)$$

For each structure M for $\sigma = \{N, s, 0\}$, $M \vDash_{\mathcal{L}_{TC}} \phi \wedge \varphi$ iff M is an ω-structure. Then, for each sentence ψ in $\mathcal{L}_{\omega}$ the equivalent sentence in $\mathcal{L}_{TC}$ is $\psi \wedge \varphi \wedge \phi$. Thus, by Proposition 4.10 we get that $\mathcal{L}_{TC} \overset{s}{\geq} \mathcal{L}_{\omega}$. □

The definition of strong inclusion prevents us from expanding our signature when looking for a translation between two logics. As a result, given a specific signature there are differences in the expressive power of the logics.

Theorem 4.14. $\mathcal{L}_{Card} \overset{s}{\ngeq} \mathcal{L}_{TC}$.

Proof. Let $\sigma = \{0, 1, +, \cdot, <\}$ be the signature of real analysis. By Tarski's theorem we have a set of sentences T in $L^1(\sigma)$ which strongly defines real closed fields. Let us define a sentence in $\mathcal{L}_{TC}$ which asserts that for every number there is a natural number larger then it:

$$\varphi := \forall x \exists y \left(x < y \wedge \left(TC_{x,y} \left(y = x + 1 \right) \right) (1, y) \right)$$

We get that $M \models T \cup \{\varphi\}$ iff M is an archimedean real closed field. Assume that $\mathcal{L}_{Card} \overset{s}{\geq} \mathcal{L}_{TC}$. Thus, there is a set of sentences Γ in $\mathcal{L}_{Card}^{\sigma}$ which defines the same class of structures as $T \cup \{\varphi\}$, i.e. Γ is satisfied by all archimedean real closed fields and only them. On the other hand, by a theorem of Cowles[14] we get that for each formula θ of $\mathcal{L}_{Card}$ above σ, there is a quantifier-free formula θ' with the same free variables as θ, such that $\theta' \leftrightarrow \theta$ holds in all models of the theory of real closed fields. Let Γ' be the set obtained by replacing each $\psi \in \Gamma$ by its quantifier-free equivalent ψ'. We now get that Γ' is satisfied by all and only archimedean real closed fields. This leads to a contradiction, since by the usual compactness argument we get that there is no first-order theory which is satisfied by only those fields. □

Theorem 4.15. *$\mathcal{L}_{TC^*} \overset{s}{\ngeq} \mathcal{L}_{Card}$ and $\mathcal{L}_{TC^*} \overset{s}{\ngeq} \mathcal{L}_{WSO}$.*

Proof. Let σ be a first-order signature with equality which contains only monadic predicates, and let φ be a formula in $L_{TC^*}(\sigma)$. For any structure M for σ and assignment v for M: if there is a natural number m such that $M, v \models \exists \bar{z_1} ... \exists \bar{z_m} \left(\bar{z_1} = \bar{s} \wedge \bar{z_m} = \bar{t} \wedge \varphi \left\{ \frac{\bar{z_1}}{\bar{x}}, \frac{\bar{z_2}}{\bar{y}} \right\} \wedge ... \wedge \varphi \left\{ \frac{\bar{z_{m-1}}}{\bar{x}}, \frac{\bar{z_m}}{\bar{y}} \right\} \right)$ where $\bar{z_1}, ..., \bar{z_m}$ are vectors of fresh variables, then $M, v \models_{\mathcal{L}_{TC^*}} \left(TC^k_{\bar{x},\bar{y}} \varphi \right) (\bar{s}, \bar{t})$. On the other hand, if $M, v \models_{\mathcal{L}_{TC^*}} \left(TC^k_{\bar{x},\bar{y}} \varphi \right) (\bar{s}, \bar{t})$ we get as a consequence of the decidability proof of monadic predicate calculus (see Dreben and Goldfarb[15]) that there is a natural number m such that:
$M, v \models \exists \bar{z_1} ... \exists \bar{z_m} \left(\bar{z_1} = \bar{s} \wedge \bar{z_m} = \bar{t} \wedge \varphi \left\{ \frac{\bar{z_1}}{\bar{x}}, \frac{\bar{z_2}}{\bar{y}} \right\} \wedge ... \wedge \varphi \left\{ \frac{\bar{z_{m-1}}}{\bar{x}}, \frac{\bar{z_m}}{\bar{y}} \right\} \right)$, where $\bar{z_1}, ..., \bar{z_m}$ are vectors of fresh variables. Therefore, for this signature σ , $\mathcal{L}_{TC^*}^{\sigma}$ is equivalent to $L^1(\sigma)$. The usual compactness argument establishes that finitude cannot be expressed in $L^1(\sigma)$. Thus, if P is any monadic predicate symbol in σ, there is no set of sentences in $\mathcal{L}_{TC^*}$ equivalent to $QxP(x)$ of $\mathcal{L}_{Card}$, or to $\neg \exists X \forall x (Xx \leftrightarrow P(x))$ of $\mathcal{L}_{WSO}$. □

Note 4.16. In addition to the results given in Theorems 4.11, 4.12, 4.13, 4.14 and 4.15, the following relations (which are not directly connected with $\mathcal{L}_{TC}$) hold as well: $\mathcal{L}_{WSO} \overset{s}{\geq} \mathcal{L}_{Card}$, $\mathcal{L}_{WSO} \overset{s}{\geq} \mathcal{L}_{NH}$, $\mathcal{L}_{WSO}, \mathcal{L}_{Card}, \mathcal{L}_{NH} \overset{s}{\geq} \mathcal{L}_{\omega}$, $\mathcal{L}_{Card} \overset{s}{\ngeq} \mathcal{L}_{WSO}$ and $\mathcal{L}_{Card} \overset{s}{\ngeq} \mathcal{L}_{NH}$. [4]

[4] Proofs can be found in [6, 16].

4.3.2 Quasi inclusion

From an examination of the cases above in which we do not have strong inclusion, it seems that they are due to the severe restriction on the signature. Therefore, a more reasonable method of comparison between the expressive power of logics is needed. In order to do so we may choose to say that two logics are equivalent if they can express the same assertions, not necessarily using the same signature. Two logics will be called equivalent if they are able to define (practically) the same classes of structures.

In this subsection our interest will be focused on infinite structures (which are the relevant structures for the goal of formalizing mathematical reasoning). Therefore two logics will be called equivalent in this subsection if they are able to define practically the same classes of infinite structures. The intuition here is that, when restricted to infinite structures, the logics under study are equivalent using this less strict type of translation.

Definition 4.17. Let $\mathcal{L}_1, \mathcal{L}_2$ be two logics. We say that $\mathcal{L}_2$ *quasi includes* $\mathcal{L}_1$ ($\mathcal{L}_2 \overset{q}{\geq} \mathcal{L}_1$) if given a first-order signature σ, there is a signature $\sigma' \supseteq \sigma$, such that every set of infinite structures definable in $\mathcal{L}_1^{\sigma}$ can be defined in $\mathcal{L}_2^{\sigma'}$. We say that $\mathcal{L}_1$ and $\mathcal{L}_2$ are *quasi equivalent* ($\mathcal{L}_2 \overset{q}{\equiv} \mathcal{L}_1$) if both $\mathcal{L}_2 \overset{q}{\geq} \mathcal{L}_1$ and $\mathcal{L}_1 \overset{q}{\geq} \mathcal{L}_2$.

Strong inclusion obviously implies quasi inclusion, thus all the positive theorems from the last subsection holds. However, here we obtain a much stronger connection between the logics of this chapter.

Theorem 4.18. *$\mathcal{L}_{WSO}$, $\mathcal{L}_{TC}$, $\mathcal{L}_{TC^*}$, $\mathcal{L}_{Card}$, $\mathcal{L}_{NH}$ and $\mathcal{L}_{\omega}$ are all quasi equivalent.*

Proof. From the strong inclusion theorems, we get that it suffices to show that $\mathcal{L}_{\omega}$ quasi includes $\mathcal{L}_{WSO}$. We show this by adding a way to code finite sets. We know that a binary relation can easily code subsets of the domain, and in general, a $(n+1)$-ary relation can easily code subsets of D^n [5]. Yet, by Cantor's theorem we know that no relation can represent every subset of the domain. However, if the domain is infinite, there is a relation which

[5] If P is a $(n+1)$-ary predicate symbol, define $P_x = \{\bar{y} \mid P(x, \bar{y})\}$. P is said to represent the collection of all sets P_x where x ranges over the domain.

represents all of its finite subsets. We will show that such a relation can be characterized in $\mathcal{L}_\omega$. For each $n > 0$ let R^{n+1} be a $(n+1)$-ary relation symbol, and let $\psi_1[R^{n+1}]$ be the formula:

$$\psi_1[R^{n+1}] := \exists x \forall \bar{y} \neg R^{n+1}(x, \bar{y}) \wedge \forall x \forall \bar{y} \exists z \forall \bar{w} \left(R^{n+1}(z, \bar{w}) \leftrightarrow \left(R^{n+1}(x, \bar{w}) \vee \bar{w} = \bar{y}\right)\right)$$

where $\bar{y}, \bar{w}$ are vectors of variables of length n, and $\bar{w} = \bar{y}$ is an abbreviation for $\bigwedge\limits_{1 \leq i \leq n} (w_i = y_i)$. The first conjunct asserts that there is a representation for the empty set in R^{n+1}, and the second conjunct asserts that if a set X is represented by R^{n+1} then, for any element $\bar{y}$, $X \cup \{\bar{y}\}$ is represented by R^{n+1} . i.e. $\psi_1[R^{n+1}]$ asserts that every finite subset of D^n is represented by R^{n+1}. It remains to ensure that *only* finite sets are represented by R^{n+1}. We introduce a binary relation K such that $K(x, y)$ entails that $N(x)$ holds, and the cardinality of $R^{n+1}_y = \{\bar{z} \mid R^{n+1}(y, \bar{z})\}$ is the natural number corresponding to x. Define $\psi_3[R^{n+1}, K]$ to be the conjunction of the assertion that $K(0, y)$ holds iff R^{n+1}_y is the empty set with the assertion that $K(s(x), y)$ holds iff there are w and $\bar{z}$ such that $K(x, w)$, $\bar{z}$ is not in R^{n+1}_w, and $R^{n+1}_y = R^{n+1}_w \cup \{\bar{z}\}$.

$$\psi_3[R^{n+1}, K] := \forall y \left(K(0, y) \leftrightarrow \forall \bar{z} \left(\neg R^{n+1}(y, \bar{z})\right)\right) \wedge$$
$$\big[K(s(x), y) \leftrightarrow \exists w \exists \bar{z} \big(K(x, w) \wedge \neg R^{n+1}(w, \bar{z}) \wedge \forall \bar{u} \big(R^{n+1}(y, \bar{u}) \leftrightarrow \big(R^{n+1}(w, \bar{u}) \vee \bar{u} = \bar{z})\big)\big)\big]$$

Define $\psi_2[K]$ to assert that for every y there is an x in N that represents the cardinality of R^{n+1}_y.

$$\psi_2[K] := \forall y \exists x \left(N(x) \wedge K(x, y)\right)$$

Now, let $\psi[R^{n+1}, K]$ be $\psi_1[R^{n+1}] \wedge \psi_2[K] \wedge \psi_3[R^{n+1}, K]$. In any ω-model of $\psi[R^{n+1}, K]$, R^{n+1} represents the set of all finite subsets D^n. Let Γ be a set of sentences in $\mathcal{L}^\sigma_{WSO}$ such that the symbols K, N and R^{n+1} for each $n > 0$ do not occur in any formula in Γ. We associate a unique fresh first-order variable x^n_X to each n-ary second-order variable X . For each sentence φ in $L^2(\sigma)$ we shall define by induction a first-order sentence φ' as follows:

- If φ is a first-order atomic formula then φ' is φ.
- If φ is a formula of the form $X^n(\bar{t})$ where X^n is a n-ary relation variable and $\bar{t}$ is a vector of n terms, then $\varphi' = R^{n+1}(x^n_X, \bar{t})$.
- $(\varphi \rightarrow \psi)'$ is $\varphi' \rightarrow \psi'$
- $(\neg\varphi)'$ is $\neg\varphi'$

- $(\forall x\varphi)'$ is $\forall x\varphi'$
- $(\forall X^n\varphi)'$ is $\forall x^n_X\varphi'$

Take $\phi_n := \bigwedge\limits_{0<i\leq n} \psi[R^{n+1}, K]$. It is easy to check that for each infinite structure M for σ , $M \models_{\mathcal{L}_{WSO}} \Gamma$ iff there is an expansion of M which satisfies $\Gamma' \cup \{\phi_k \mid k \in \mathbb{N}\}$ in $\mathcal{L}_\omega$. □

Proposition 4.19. *The simple Lowenhiem-Skolem theorem, which states that if a theory has an infinite model then it has a countable one, holds for the logics of this chapter.*

Proof. A routine check of the usual proof will verify that if the original structure M is an ω-model, then there is a countable sub-structure M' which satisfies exactly the same sentences as M and is also an ω-model. Thus, the downward Lowenheim-Skolem theorem holds for $\mathcal{L}_\omega$. Let $\mathcal{L}$ be one of the logics $\mathcal{L}_{WSO}$, $\mathcal{L}_{TC}$, $\mathcal{L}_{TC^*}$, $\mathcal{L}_{Card}$ or $\mathcal{L}_{NH}$, σ a first-order signature and Γ a set of sentences in $\mathcal{L}^\sigma$. Let M be a structure for σ such that $M \models_{\mathcal{L}} \Gamma$. Since $\mathcal{L}_\omega$ is quasi equivalent to $\mathcal{L}$, there is a set of sentences Γ' in $\mathcal{L}^\sigma_\omega$ such that $M \models_{\mathcal{L}_\omega} \Gamma'$. It follows from the downward Lowenheim-Skolem theorem for $\mathcal{L}_\omega$ that there is a countable ω-structure M' such that $M' \models_{\mathcal{L}_\omega} \Gamma'$, and again by the quasi equivalence we get that $M' \models_{\mathcal{L}} \Gamma$. □

4.3.3 Inclusion

In this subsection we present the same form of comparison between logics presented in the last subsection only without the restriction to infinite structures. We say that two logics are equivalent if they are able to define the same classes of structures without preserving signature.

Definition 4.20. Let $\mathcal{L}_1, \mathcal{L}_2$ be two logics. We say that $\mathcal{L}_2$ *includes* $\mathcal{L}_1$ ($\mathcal{L}_2 \geq \mathcal{L}_1$) if given a first-order signature σ, there is a signature $\sigma' \supseteq \sigma$, such that every set of structures definable in $\mathcal{L}^\sigma_1$ can be defined in $\mathcal{L}^{\sigma'}_2$. We say that $\mathcal{L}_1$ and $\mathcal{L}_2$ are *equivalent* ($\mathcal{L}_2 \equiv \mathcal{L}_2$) if both $\mathcal{L}_2 \geq \mathcal{L}_1$ and $\mathcal{L}_1 \geq \mathcal{L}_2$.

We prove a series of theorems from which it will follow that almost all of the logics of this chapter are equivalent. In order to prove that $\mathcal{L}_{TC^*} \equiv \mathcal{L}_{NH}$ we will need the following 2 lemmas.[6]

[6] The following results appear in [19, 16].

Lemma 4.21. *Let an equality bound narrow Henkin quantifier be a narrow Henkin quantifier whose second-order form is equivalent to the following formula:* $\exists f \forall \bar{x} \forall \bar{y} \psi \left(\bar{x}, \bar{y}, f\left(\bar{x}\right), f\left(\bar{y}\right)\right)$*. Narrow Henkin quantifiers and equality bound narrow Henkin quantifiers can be expressed in terms of one another.*

Proof. An equality bound narrow Henkin quantifier is equivalent to the following narrow Henkin quantifier:

$$\left(\begin{array}{cc} \forall \bar{x} & \exists \alpha \\ \forall \bar{y} & \exists \beta \end{array}\right) \left(\bar{x} = \bar{y} \rightarrow \alpha = \beta\right) \wedge \psi \left(\bar{x}, \bar{y}, \alpha, \beta\right)$$

For the other direction, the second-order form semantics of a general narrow Henkin quantifier is:

$$\Psi := \exists f \exists g \forall \bar{x} \forall \bar{y} \psi \left(\bar{x}, \bar{y}, f\left(\bar{x}\right), g\left(\bar{y}\right)\right)$$

Now, we can define an equality bound narrow Henkin quantifier:

$$\Phi := \exists h \forall \bar{x} \gamma \forall \bar{y} \delta \left(\gamma = 1 \wedge \delta = 0 \rightarrow \psi \left(\bar{x}, \bar{y}, h\left(\bar{x}, \gamma\right), h\left(\bar{y}, \delta\right)\right)\right)$$

We can now see that Φ implies Ψ using the following definitions:

$$f\left(\bar{x}\right) = h\left(\bar{x}, 1\right) \quad , \quad f\left(\bar{y}\right) = h\left(\bar{y}, 0\right)$$

□

Lemma 4.22. *Let* $(X, \leq)$ *be a pre-ordered set, and let* $i : X \rightarrow X$ *be an order-reversing function such that for every* $x \in X$ *,* $i\left(i\left(x\right)\right) = x$ *and never both* $i\left(x\right) \leq x$ *and* $x \leq i\left(x\right)$*. Then there is a* $T \subseteq X$ *, closed upwards for* $\leq$*, which contains exactly one of* x *and* $i\left(x\right)$ *for each* $x \in X$ *.*

Proof. Let S consist of all sets $A \subseteq X$ which are closed upwards for $\leq$, and contain at most one of x and $i\left(x\right)$ for each $x \in X$. S is not empty since it contains at least the empty set. S is partially ordered by set inclusion. Take any totally ordered subset C of S. It is easy to see that the set $A' = \bigcup\limits_{A \in C} A$ is an upper bound of C. By Zorn's lemma we get that there is a maximal element in S, say T. Assume that there is $y \in X$ such that both $y \notin T$ and $i\left(y\right) \notin T$. WLOG assume that $y \nleq i\left(y\right)$. Define $T' = T \cup \{x \in X \mid y \leq x\}$. T' is obviously closed upwards for $\leq$. Assume that there exists $z \in X$ such that both $z \in T'$ and $i\left(z\right) \in T'$. z and $i\left(z\right)$ cannot be both in T. If $z \notin T$, then $y \leq z$. Thus, $i(z) \leq i(y)$. But since $i(z) \in T'$ and T' is closed upwards for $\leq$, we get that $i(y) \in T'$. $i(y) \notin T$ implies that $y \leq i(y)$, which contradicts the assumption that $y \nleq i(y)$. Since $i(i(x)) = x$ for every element x, similar arguments lead to a contradiction in the case that $i(z) \notin T$. Thus, T' contains at most one of x and $i(x)$ for each $x \in X$. Since $T \subset T'$ (because $y \in T' \setminus T$), T' contradicts the maximality of T. Therefore, there is no $y \in X$ such that both $y \notin T$ and $i(y) \notin T$. Hence, T has all the required properties. □

Theorem 4.23. $\mathcal{L}_{TC^*} \equiv \mathcal{L}_{NH}$

Proof. Let us start by showing that $\mathcal{L}_{NH}$ includes $\mathcal{L}_{TC^*}$. We know that the following holds:

$$\left(TC^k_{\bar{x},\bar{y}}\varphi\right)(\bar{s},\bar{t}) \equiv$$
$$\exists\bar{z_0}...\exists\bar{z_n}\left(\bar{z_0} = \bar{s} \land \bar{z_n} = \bar{t} \land \varphi\left\{\frac{\bar{z_0}}{\bar{x}}, \frac{\bar{z_1}}{\bar{y}}\right\} \land ... \land \varphi\left\{\frac{\bar{z_{n-1}}}{\bar{x}}, \frac{\bar{z_n}}{\bar{y}}\right\}\right)$$

If we add two constants symbols: 0 and 1 we get the equivalent second-order formula:

$$\neg\exists f\left(f(\bar{s}) = 1 \land f(\bar{t}) = 0 \land \forall\bar{x}\forall\bar{y}\left(f(\bar{x}) = 1 \land \varphi(\bar{x},\bar{y}) \rightarrow f(\bar{y}) = 1\right)\right)$$

This second-order formula is equivalent to the following formula in $\mathcal{L}_{NH}$:

$$\neg\begin{pmatrix} \forall\bar{x} & \exists\alpha \\ \forall\bar{y} & \exists\beta \end{pmatrix}((\bar{x} = \bar{y} \rightarrow \alpha = \beta) \land (\bar{x} = \bar{s} \rightarrow \alpha = 1) \land$$
$$(\bar{x} = \bar{t} \rightarrow \alpha = 0) \land (\alpha = 1 \land \varphi(\bar{x},\bar{y}) \rightarrow \beta = 1))$$

For the converse, we get from Lemma 4.21 and the equivalence between $\mathcal{L}_{TC^*}$ and $\mathcal{L}_{RTC^*}$ (see Proposition 3.6) that it suffices to express a general formula with an equality bound narrow Henkin quantifier in terms of the reflexive transitive closure operator. The second-order semantics of an equality bound narrow Henkin quantifier is:

$$\exists f\forall\bar{x}\forall\bar{y}\psi\left(\bar{x}, f(\bar{x}), \bar{y}, f(\bar{y})\right)$$

where f in a boolean function.

Let us assume that such a function exists. Let us also assume that, for certain $\bar{x}, \bar{y}, \alpha, \beta$, $\neg\psi(\bar{x}, \alpha, \bar{y}, 1-\beta)$ is true. We then get both that $\psi(\bar{x}, f(\bar{x}), \bar{y}, f(\bar{y}))$ is true, and that $\psi(\bar{x}, f(\bar{x}), \bar{y}, f(\bar{y}))$ is false. Thus, if

$f(\bar{x})$ is assigned the value α, then $f(\bar{y})$ must be assigned the value β. The same result is obtained in the case that $\neg\psi(\bar{y}, 1-\beta, \bar{x}, \alpha)$ is true.
Let us now define:

$$\phi(\bar{x}\hat{}\alpha, \bar{y}\hat{}\beta) := \neg\psi(\bar{x}, \alpha, \bar{y}, 1-\beta) \vee \neg\psi(\bar{y}, 1-\beta, \bar{x}, \alpha)$$

where $\hat{}$ stands for the concatenation function.

When $\phi(\bar{x}\hat{}\alpha, \bar{y}\hat{}\beta)$ holds, the value α for $f(\bar{x})$ "forces" the value β for $f(\bar{y})$. Since this "forcing" is a reflexive transitive relation on tuples of $k+1$ elements $\bar{x}\hat{}\alpha$, we see that $\bar{s}$ forces $\bar{t}$ whenever $\left(RTC^{k+1}_{\bar{x}\alpha,\bar{y}\beta}\phi\right)(\bar{s}, \bar{t})$ holds. In particular, if for a certain $\bar{a}$, $\left(RTC^{k+1}_{\bar{x}\alpha,\bar{y}\beta}\phi\right)(\bar{a}\hat{}\gamma, \bar{a}\hat{}(1-\gamma))$ holds, then $f(\bar{a})$ cannot be assigned the value γ, or we will get a contradiction. Thus, a necessary condition for such an assignment f to exist is :

$$\neg\exists\bar{a}\exists\gamma\left[\left(RTC^{k+1}_{\bar{x}\alpha,\bar{y}\beta}\phi\right)(\bar{a}\hat{}\gamma, \bar{a}\hat{}(1-\gamma)) \wedge \left(RTC^{k+1}_{\bar{x}\alpha,\bar{y}\beta}\phi\right)(\bar{a}\hat{}(1-\gamma), \bar{a}\hat{}\gamma)\right] \quad (4.1)$$

This sentence states that $f(\bar{a})$ cannot be assigned both values γ and $(1-\gamma)$, which is of course not possible since f is taken to be a boolean function.

We now have to show that this condition is also sufficient. As a consequence of the definition of the operator RTC, $\left(RTC^{k+1}_{\bar{x}\hat{}\alpha,\bar{y}\hat{}\beta}\phi\right)$ defines a pre-ordering (a relation $\leq$) on the tuples $\bar{x}\hat{}\alpha$. Moreover, if the structure satisfies the necessary condition (4.1) then there cannot be both $\bar{x}\hat{}\alpha \leq \bar{x}\hat{}(1-\alpha)$ and $\bar{x}\hat{}(1-\alpha) \leq \bar{x}\hat{}\alpha$. Now let us apply Lemma 4.22 to the set of tuples $\bar{x}\hat{}\alpha$ pre-ordered according to $\left(RTC^{k+1}_{\bar{x}\hat{}\alpha,\bar{y}\hat{}\beta}\phi\right)$, with $i(\bar{x}\hat{}\alpha) = \bar{x}\hat{}(1-\alpha)$. It is easy to see that i is an order-reversing function since $\left(RTC^{k+1}_{\bar{x}\hat{}\alpha,\bar{y}\hat{}\beta}\phi\right)$ is invariant when interchanging $\bar{x}$ and $\bar{y}$, and replacing α with $(1-\beta)$ and β with $(1-\alpha)$. The last assumption of the lemma is met due to the necessary condition (4.1) . Then, we get a set T as guaranteed by the lemma.

Now define:

$$f(x) = \begin{cases} 1 & \bar{x}\hat{}1 \in T \\ 0 & otherwise \end{cases}$$

Thus, we always have $\bar{x}\hat{}f(x) \in T$ and $\bar{x}\hat{}(1-f(x)) \notin T$.

It remains to show that this assignment f has the desired property that $\psi(\bar{x}, f(\bar{x}), \bar{y}, f(\bar{y}))$ holds for all $\bar{x}$ and $\bar{y}$. To see this, suppose that for certain $\bar{x}$ and $\bar{y}$, $\neg\psi(\bar{x}, f(\bar{x}), \bar{y}, f(\bar{y}))$ holds. Then $\phi(\bar{x}\hat{}f(x), \bar{y}\hat{}(1-f(y)))$ holds, and since ϕ implies $\left(RTC^{k+1}_{\bar{x}\alpha,\bar{y}\beta}\phi\right)$, we have that $\bar{x}\hat{}f(x) \leq \bar{y}\hat{}(1-f(y))$. But $\bar{x}\hat{}f(x) \in T$ and $\bar{y}\hat{}(1-f(y)) \notin T$, and since T is closed upwards, we get a contradiction. $\square$

Theorem 4.24. $\mathcal{L}_{TC} \geq \mathcal{L}_{Card}$

Proof. Let ψ be a sentence in $\mathcal{L}^{\sigma}_{Card}$ and let f be a unary function symbol not in σ. Now let ϕ be a first-order sentence which asserts that f is $1-1$ and there is an element not in the range of f. For any structure M for σ, the domain of M is infinite iff there is an expansion of M which satisfies ϕ. Next let φ be a first-order sentence which asserts that there is an x such that everything (including x) is an ancestor of $f(x)$ under f . For any structure M for σ, the domain of M is finite iff there is an expansion of M which satisfies φ.

Since $\mathcal{L}_{TC}$ quasi includes $\mathcal{L}_{Card}$ (see Theorem 4.18), there is a signature $\sigma \subseteq \sigma'$ and a sentence of $\mathcal{L}^{\sigma'}_{TC}$, ψ', such that for every infinite structure M, $M \models_{\mathcal{L}_{Card}} \psi$ iff there is an expansion of M ,M', such that $M' \models_{\mathcal{L}_{TC}} \psi'$. Let ψ'' be the sentence obtained by replacing each sub-formula of ψ of the form $Q_0 x\theta$ with a logical contradiction. If M is infinite then it satisfies ψ iff there is an expansion of M that satisfies ψ', and if M is finite then M satisfies ψ iff there is an expansion of M that satisfies ψ''. Therefore, we get that for any structure M , M satisfies ψ iff there is an expansion of M which satisfies $(\phi \wedge \psi') \vee (\varphi \wedge \psi'')$. □

Theorem 4.25. $\mathcal{L}_{Card} \geq \mathcal{L}_{TC^*}$

Proof. That $\mathcal{L}_{Card}$ includes $\mathcal{L}_{TC^*}$ is done by arguments similar to the ones used in the last proof. Let ψ be a sentence in $\mathcal{L}^{\sigma}_{TC^*}$. Since $\mathcal{L}_{Card}$ quasi includes $\mathcal{L}_{TC^*}$ (Theorem 4.18), there is a signature $\sigma \subseteq \sigma'$ and a sentence of $\mathcal{L}^{\sigma'}_{Card}$, ψ', such that for every infinite structure M, $M \models_{\mathcal{L}_{TC^*}} \psi$ iff there is an expansion of M ,M', such that $M' \models_{\mathcal{L}_{Card}} \psi'$. Now, we show that there is a signature $\sigma \subseteq \sigma'$ and a sentence ψ'' in $L^1(\sigma')$ such that, for each finite structure M, $M \models_{\mathcal{L}_{TC}} \psi$ iff there is an expansion M' of M such that $M' \vDash_{L^1(\sigma')} \psi''$. This can be accomplished by introducing terminology for a linear order, which allows objects in the domain to play the role of (some) natural numbers. Then for any formula $\varphi(\bar{x}, \bar{y})$ where $\bar{x}, \bar{y}$ are vectors of length k of distinct variables, we can formulate a formula $\varphi'(\bar{x}, \bar{y}, z)$ which states that " $\bar{y}$ is an ancestor of $\bar{x}$ under φ by a chain whose length corresponds to z ". Thus, let ψ'' be the sentence obtained by replacing each sub-formula of ψ of the form $(TC_{\bar{x},\bar{y}}\varphi)(\bar{s}, \bar{t})$ with $\exists z (\varphi'(\bar{s}, \bar{t}, z))$. Therefore, again we get that for any structure M , M satisfies ψ iff there is an expansion of M which satisfies $(\phi \wedge \psi') \vee (\varphi \wedge \psi'')$, where ϕ, φ are as in the proof of Theorem 4.24. □

From Theorems 4.23, 4.24 and 4.25 we get the following.

Theorem 4.26. *$\mathcal{L}_{TC}$, $\mathcal{L}_{TC^*}$, $\mathcal{L}_{Card}$ and $\mathcal{L}_{NH}$ are all equivalent.*

Remark 4.27. The question whether $\mathcal{L}_{WSO}$ is equivalent to the logics mentioned in Theorem 4.26 is equivalent to a long-standing open problem in complexity theory. The crucial proposition is that for every sentence ψ of $\mathcal{L}_{WSO}$, there is a sentence ψ' of $\mathcal{L}_{Card}$ such that, for each *finite* structure M, $M \vDash_{\mathcal{L}_{WSO}} \psi$ iff there is an expansion M' of M such that $M' \vDash_{\mathcal{L}_{Card}} \psi'$. When attention is restricted to finite structures, $\mathcal{L}_{WSO}$ is the same as second-order logic and each sentence of $\mathcal{L}_{Card}$ is equivalent to a first-order sentence. Thus the proposition turns to: For every sentence ψ of the second-order $L^2(\sigma)$, there is a set $\sigma' \supseteq \sigma$ and a sentence ψ' in $L^1(\sigma')$ such that for each finite structure M, $M \vDash_{L^2(\sigma)} \psi$ iff there is an expansion M' of M such that $M' \vDash_{L^1(\sigma')} \psi'$. The proposition can be reduced further to the point where it becomes equivalent to the open problem in complexity theory concerning whether the properties of finite structures recognized by NP algorithms include the full polynomial-time hierarchy.

4.4 Conclusion

Let us summarize the main results of this chapter. The most important conclusion is that by adding to FOL each of the basic concepts investigated in this chapter (the ancestral, new types of quantifiers or relation variables, etc.) we obtain logics which are equivalent in the sense that any class of infinite structures definable by one of them can be defined by any of the others. It follows that these logics provide a natural intermediate level between FOL and SOL.

In comparison to FOL, all the logics of this chapter have the advantage that they are not compact (the natural numbers can be characterized up to isomorphism in all of them). On the other hand, there is no complete deductive system which is sound for any of these logics. Unlike the downward Lowenheim-Skolem theorem, the upwards upward Lowenheim-Skolem theorem fails for these logics.

Though more expressive than FOL, the logics of this chapter do not offer all the wealth of SOL. Thus, it follows from Proposition 4.19 that the real numbers cannot be characterized up to isomorphism in either of the logics (while they can be characterized in SOL.). The same is true for the notion of well-ordering.

5 Formal proof systems for ancestral logic

5.1 Previous proof systems for ancestral logic

Ideally, we would like to have a consistent, sound and complete axiomatic system for ancestral logic. However, from Corollary 4.6 it follows that there is no such system. Thus, one should instead look for a useful and effective *partial* axiomatic systems which are still adequate for formalizing mathematical reasoning. In [3, 4, 5] R.M. Martin and J. Myhill suggested two equivalent systems for this purpose. Those systems were Hilbert-style systems. Accordingly, we start this chapter by presenting an Hilbert-style system for ancestral logic, RTC_H, which is a variation on those systems for the RTC operator.

Definition 5.1 (*The system RTC_H*)**.**
The *system RTC_H* is defined by adding to the basic Hilbert's system for first-order logic the following three axioms for the RTC operator*:*

$$RTC_{x,y}\varphi\left(s,s\right) \tag{5.1}$$

$$\varphi\left\{\frac{s}{x},\frac{r}{y}\right\}\wedge\left(RTC_{x,y}\varphi\right)\left(r,t\right)\rightarrow\left(RTC_{x,y}\varphi\right)\left(s,t\right) \tag{5.2}$$

$$\forall x\forall y\left(\psi\left(x\right)\wedge\varphi\left(x,y\right)\rightarrow\psi\left\{\frac{y}{x}\right\}\right)\wedge\psi\left\{\frac{s}{x}\right\}\wedge\left(RTC_{x,y}\varphi\right)\left(s,t\right)\rightarrow\psi\left\{\frac{t}{x}\right\} \tag{5.3}$$

Remark 5.2. It would suffice to take the above axioms as sentences. For example instead of 5.2, to take

$$\forall x\forall y\forall z\left(\varphi\wedge\left(RTC_{x,y}\varphi\right)\left(y,z\right)\rightarrow\left(RTC_{x,y}\varphi\right)\left(x,z\right)\right)$$

Martin and Myhill both took as primitive the concept of the reflexive transitive closure operator. An Hilbert-style system TC_H for the non-reflexive transitive closure operator can be given by slightly modifying the axioms of RTC_H. The only difference to be made in Axioms 5.2 and 5.3 is replacing the RTC operator with the TC operator. Axiom 5.1 is to be replaced by the axiom $\varphi\left\{\frac{s}{x},\frac{t}{y}\right\}\rightarrow\left(TC_{x,y}\varphi\right)\left(s,t\right)$.

Hilbert-style systems are not very useful from a proof-theoretic point of view. Thus we next present an equivalent proof system for ancestral logic of a more adequate type - Gentzen-style system.

5.2 Gentzen-style systems

In [8], Gentzen introduced a new concept of formal proof systems which instead of using formulas as syntactic entities, use a new data-structure called sequents.

Definition 5.3. A *sequent* is an expression of the form $\Gamma \Rightarrow \Delta$, Where Γ, Δ denote finite (possibly empty) multisets of formulas[7]. Γ is called the antecedent of the sequent, and Δ is called the succedent of the sequent.

The intuitive meaning of a sequent of the form $\varphi_1, ..., \varphi_m \Rightarrow \psi_1, ..., \psi_n$ is $\varphi_1 \wedge ... \wedge \varphi_m \rightarrow \psi_1 \vee ... \vee \psi_n$. If the antecedent is empty, the sequent reduces to the formula $\psi_1 \vee ..., \vee \psi$. If the succedent is empty, the sequents reduces to $\neg (\varphi_1 \wedge ... \wedge \varphi_m)$. The empty sequent means a contradiction.

Definition 5.4. A sequential *inference step* is an expression of the form

$$\frac{S_1 ... S_n}{S}$$

Where $S_1, ..., S_n$ and S are sequents. $S_1, ..., S_n$ are called the upper sequents or the premises, and S is called the lower sequent or the conclusion of the inference step.

Intuitively, the meaning of an inference step is that when the upper sequents are asserted, we can infer the lower sequent.

Definition 5.5. A formal *proof system* comprises of a set of sequents which are taken as axioms and a set of rules for making inference steps, called inference rules.

Definition 5.6. Let G be a Gentzen-style system and S a set of sequents. A *derivation*, or a proof-figure, from S in G is a finite tree of sequents such that:

- the topmost sequents of the derivation are axioms of G or elements of S.
- Each sequent (apart from the axioms) is obtained from previous sequents by one of the inference rules of G.

[7] Gentzen originally took Γ, Δ to be *sequences* of formulas.

The lower sequent in a derivation P is called the end-sequent of P. A proof with end-sequent s is called a proof of s.

Definition 5.7. Let G be a Gentzen-style system, S a set of sequents and s a sequent. s is said to be *provable* from S in a system G, denoted by $S \vdash_G s$, if there exists a derivation in G of s from S. The sequents of S are called assumptions. We say that s is a *theorem* of the system G if it is derivable from the empty sequent.

It is a standard notation to abbreviate part of a proof by "$\vdots$". Thus, for example,

$$\begin{array}{c}\vdots\, P\\ S\end{array}$$

denotes a proof P ending with the sequent S.

Gentzen introduced two sequential calculi: $\mathcal{LK}$ for classical logic and $\mathcal{LJ}$ for intuitionistic logic. $\mathcal{LK}$ was proved to be sound and complete with respect to classical logic, and $\mathcal{LJ}$ with respect to intuitionistic logic. Since our interest is focused on classical logics with equality we shall present the system $\mathcal{LK}_=$. In the system for each connective or quantifier there are two types of inference rules: introduction on the left, which introduce the connective or quantifier in the antecedent, and introduction on the right, which does the same in the succedent.

The letters Γ, Δ, Θ represents finite (possibly empty) multisets of formulas, φ and ψ arbitrary formulas, x and y variables and s and t terms. For convenience, we shall denote a sequent of the form $\Gamma \Rightarrow \{\varphi\}$ by $\Gamma \Rightarrow \varphi$, and employ other standard abbreviations such as Γ, Δ instead of $\Gamma \cup \Delta$.

Definition 5.8 (*The system $\mathcal{LK}_=$*)**.**

Axioms:

- Logical axiom:

$$\varphi \Rightarrow \varphi$$

- Equality axiom:

$$\Rightarrow s = s$$

Inference rules:

- Structural inference rules:
  - *Weakening*:

$$\frac{\Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta}\,(wkL) \quad \frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \varphi}\,(wkR)$$

  - *Contraction:*

$$\frac{\varphi, \varphi, \Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta}\,(cntL) \quad \frac{\Gamma \Rightarrow \Delta, \varphi, \varphi}{\Gamma \Rightarrow \Delta, \varphi}\,(cntR)$$

  - *Cut:*

$$\frac{\Gamma \Rightarrow \Delta, \varphi \quad \varphi, \Theta \Rightarrow \Lambda}{\Gamma, \Theta \Rightarrow \Delta, \Lambda}\,(cut)$$

  - *substitution of terms for free variables*[8]*:*

$$\frac{\Gamma \Rightarrow \Delta}{\Gamma \left\{\frac{\vec{s}}{\vec{x}}\right\} \Rightarrow \Delta \left\{\frac{\vec{s}}{\vec{x}}\right\}}\,(sub)$$

    where $\vec{s}$ is free for $\vec{x}$ in all formulas in $\Gamma \cup \Delta$.

- Equality inference rule:

$$\frac{\Gamma \Rightarrow \Delta, s = t \quad \Theta \Rightarrow \Lambda, \varphi\left\{\frac{s}{x}\right\}}{\Gamma, \Theta \Rightarrow \Delta, \Lambda, \varphi\left\{\frac{t}{x}\right\}}\,(eq1) \quad \frac{\Gamma \Rightarrow \Delta, s = t \quad \Theta \Rightarrow \Lambda, \varphi\left\{\frac{t}{x}\right\}}{\Gamma, \Theta \Rightarrow \Delta, \Lambda, \varphi\left\{\frac{s}{x}\right\}}\,(eq2)$$

  where s, t are free for x in φ.

- Operational inference rules:
  - *Conjunction:*

$$\frac{\Gamma \Rightarrow \Delta, \varphi \quad \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi}\,(\wedge R) \quad \frac{\varphi, \Gamma \Rightarrow \Delta}{\varphi \wedge \psi, \Gamma \Rightarrow \Delta}\,(\wedge L_1) \quad \frac{\psi, \Gamma \Rightarrow \Delta}{\varphi \wedge \psi, \Gamma \Rightarrow \Delta}\,(\wedge L_2)$$

[8] *This rule was not taken as a logical rule in the original $\mathcal{LK}$ system in [8].*

- *Disjunction:*

$$\frac{\varphi, \Gamma \Rightarrow \Delta \quad \psi, \Gamma \Rightarrow \Delta}{\varphi \vee \psi, \Gamma \Rightarrow \Delta}(\vee L) \quad \frac{\Gamma \Rightarrow \Delta, \varphi}{\Gamma \Rightarrow \Delta, \varphi \vee \psi}(\vee R_1) \quad \frac{\Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \vee \psi}(\vee R_2)$$

- *Implication:*

$$\frac{\Gamma \Rightarrow \Delta, \varphi \quad \psi, \Gamma \Rightarrow \Delta}{\varphi \to \psi, \Gamma \Rightarrow \Delta}(\to L) \quad \frac{\varphi, \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \to \psi}(\to R)$$

- *Negation:*

$$\frac{\Gamma \Rightarrow \Delta, \varphi}{\neg\varphi, \Gamma \Rightarrow \Delta}(\neg L) \quad \frac{\varphi, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg\varphi}(\neg R)$$

- *Universal quantifier:*

$$\frac{\varphi\left\{\frac{t}{x}\right\}, \Gamma \Rightarrow \Delta}{\forall x\varphi, \Gamma \Rightarrow \Delta}(\forall L) \quad \frac{\Gamma \Rightarrow \Delta, \varphi\left\{\frac{y}{x}\right\}}{\Gamma \Rightarrow \Delta, \forall x\varphi}(\forall R)$$

  where y and t are free for substitution instead of x in φ, and y does not occur free in $\Gamma \cup \Delta \cup \{\forall x\varphi\}$.

- *Existential quantifier:*

$$\frac{\varphi\left\{\frac{y}{x}\right\}, \Gamma \Rightarrow \Delta}{\exists x\varphi, \Gamma \Rightarrow \Delta}(\exists L) \quad \frac{\Gamma \Rightarrow \Delta, \varphi\left\{\frac{t}{x}\right\}}{\Gamma \Rightarrow \Delta, \exists x\varphi}(\exists R)$$

  Where y and t are free for substitution instead of x in φ, and y does not occur free in $\Gamma \cup \Delta \cup \{\exists x\varphi\}$.

Remark 5.9. In the rest of this chapter we will use several times the following derivable rule of $\mathcal{LK}_=$:

$$\frac{\Gamma \Rightarrow \Delta, \varphi\left\{\frac{s}{x}\right\}}{\Gamma, s = t \Rightarrow \Delta, \varphi\left\{\frac{t}{x}\right\}}(eq3)$$

Remark 5.10. In the proofs in the rest of this chapter we will not distinguish between the sequents:

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

since each of them is provable prom the other. Notice that deriving the second sequent from the first one involves the cut rule.

The same goes for the sequents:

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

Gentzen proved a version of the cut-elimination theorem for $\mathcal{LK}_=$, which states that whenever there exists a proof (from no assumptions) of some sequent in $\mathcal{LK}_=$, there exists a proof of the same sequent without any essential cuts, i.e. cuts can only occur on formulas of the form $s = t$. This is a crucial property in any reasonable Gentzen-style system since it usually has many important corollaries, for example the sub-formula property. The original proof of the cut-elimination theorem was done by case analysis on derivation ending with an application of the cut rule. In the present chapter, $\mathcal{LK}_=$ is extended to a system for ancestral logic and similar methods are used in order to investigate the system's proof-theoretic properties.

5.3 Gentzen-style system for ancestral logic

In this section two Gentzen-style systems for ancestral logic are presented, one for the reflexive transitive closure operator and one for the non-reflexive one. The properties of the systems and the difference between them are investigated.The first system, for the TC operator, is based on the one suggested in [1].

Definition 5.11 (*The system TC_G*).
The *system TC_G* is defined by adding to the basic Gentzen's system for first-order logic $\mathcal{LK}_=$ the following inference rules for the TC operator*:*

$$\frac{\Gamma \Rightarrow \Delta, \varphi\left\{\frac{s}{x}, \frac{t}{y}\right\}}{\Gamma \Rightarrow \Delta, \left(TC_{x,y}\varphi\right)(s,t)} \tag{5.4}$$

$$\frac{\Gamma \Rightarrow \Delta, \left(TC_{x,y}\varphi\right)(s,r) \quad \Gamma \Rightarrow \Delta, \left(TC_{x,y}\varphi\right)(r,t)}{\Gamma \Rightarrow \Delta, \left(TC_{x,y}\varphi\right)(s,t)} \tag{5.5}$$

$$\frac{\Gamma, \psi(x), \varphi(x,y) \Rightarrow \Delta, \psi\left\{\frac{y}{x}\right\}}{\Gamma, \psi\left\{\frac{s}{x}\right\}, \left(TC_{x,y}\varphi\right)(s,t) \Rightarrow \Delta, \psi\left\{\frac{t}{x}\right\}} \tag{5.6}$$

In all three rules we assume that the terms which are substituted are free for substitution and that no forbidden capturing occurs. Rules 5.4 and 5.5 are introduction rules on the right. Rule 5.6 is an introduction rule on the left. The restrictions in applying this rule are that x should not occur free in Γ and Δ, and y should not occur free in Γ, Δ and ψ. This rule is a generalized induction principle which says that if t is a φ-descendant of s, then if s has some property which is passed down from one object to another if they are φ-related, then t also has that property.

Next is a definition of a Gentzen-style system for PA.

Definition 5.12 (*The system PA_G*)**.**
The *system PA_G* is obtained from $\mathcal{LK}_=$ by adding the following:

$$s(x) = 0 \Rightarrow$$
$$s(x) = s(y) \Rightarrow x = y$$

are axioms in PA_G. If φ is one of the axioms 2.3, 2.4, 2.5 or 2.6 of PA, then $\Rightarrow \hat{\varphi}$ is an axiom in PA_G[9]. The following inference rule is also added:

$$\frac{\Gamma, \psi \Rightarrow \Delta, \psi\left\{\frac{S(x)}{x}\right\}}{\Gamma, \psi\left\{\frac{0}{x}\right\} \Rightarrow \Delta, \psi\left\{\frac{t}{x}\right\}} \tag{5.7}$$

Note. Gentzen Proved that PA_G is equivalent to PA.

In TC_G augmented by Axiom 3.1 (see Theorem 3.13), the induction principle 5.6 entails all instances of the ordinary first-order induction schema. To see this assume that $\sigma \supseteq \{0, S\}$, and take φ to be $S(x) = y$ and s to be 0. We then obtain:

$$\frac{\Gamma, \psi(x), S(x) = y \Rightarrow \Delta, \psi\left\{\frac{y}{x}\right\}}{\Gamma, \psi\left\{\frac{0}{x}\right\}, (TC_{x,y}(S(x) = y))(0, t) \Rightarrow \Delta, \psi\left\{\frac{t}{x}\right\}} \tag{5.8}$$

[9] $\hat{\varphi}$ denotes the matrix of φ (see 2.3).

By using the first-order rules for $=$ and the cut rule, and by abbreviating $(TC_{x,y}(S(x)=y))(0,t)$ by $0<t$, we get:

$$\frac{\Gamma, \psi(x) \Rightarrow \Delta, \psi\left\{\frac{S(x)}{x}\right\}}{\Gamma, \psi\left\{\frac{0}{x}\right\}, 0<t \Rightarrow \Delta, \psi\left\{\frac{t}{x}\right\}} \tag{5.9}$$

Since the sequent $\Gamma, \psi\left\{\frac{0}{x}\right\}, 0=t \Rightarrow \Delta, \psi\left\{\frac{t}{x}\right\}$ is obviously valid in TC_G we have:

$$\frac{\Gamma, \psi(x) \Rightarrow \Delta, \psi\left\{\frac{S(x)}{x}\right\}}{\Gamma, \psi\left\{\frac{0}{x}\right\}, 0=t \vee 0<t \Rightarrow \Delta, \psi\left\{\frac{t}{x}\right\}} \tag{5.10}$$

It follows that in any system in which $\Rightarrow 0=t \vee 0<t$ (which is a direct consequence of 3.1) is provable we can derive the induction rule in PA_G, Rule 5.7.

An important observation should be made regarding the choice of rules in TC_G. Instead of Rule 5.5 we could have chosen each of the following rules:

$$\frac{\Gamma \Rightarrow \Delta, (TC_{x,y}\varphi)(s,r) \quad \Gamma \Rightarrow \Delta, \varphi\left\{\frac{r}{x}, \frac{t}{y}\right\}}{\Gamma \Rightarrow \Delta, (TC_{x,y}\varphi)(s,t)} \tag{5.11}$$

$$\frac{\Gamma \Rightarrow \Delta, \varphi\left\{\frac{s}{x}, \frac{r}{y}\right\} \quad \Gamma \Rightarrow \Delta, (TC_{x,y}\varphi)(r,t)}{\Gamma \Rightarrow \Delta, (TC_{x,y}\varphi)(s,t)} \tag{5.12}$$

Obviously 5.11 and 5.12 are derivable in TC_G. For example, here is a proof of 5.12 in TC_G. (The proof of 5.11 is analogous.)

$$\dfrac{\dfrac{\Gamma \Rightarrow \Delta, \varphi\left\{\frac{s}{x}, \frac{r}{y}\right\}}{\Gamma \Rightarrow \Delta, (TC_{x,y}\varphi)(s,r)}\,(5.4) \quad \Gamma \Rightarrow \Delta, (TC_{x,y}\varphi)(r,t)}{\Gamma \Rightarrow \Delta, (TC_{x,y}\varphi)(s,t)}\,(5.5)$$

The converse is also true; i.e. Rule 5.5 is derivable from Rules 5.14, 5.6 and either 5.11 or 5.12. To show this, here is a proof of 5.5 in TC'_G which is the system obtained from TC_G by replacing Rule 5.5 with 5.11. For convenience, we omit the context Γ, Δ from the sequents in the following proof.

$$\dfrac{\Rightarrow (TC_{x,y}\varphi)(r,t) \qquad \dfrac{\Rightarrow (TC_{x,y}\varphi)(s,r) \qquad \dfrac{\dfrac{(TC_{x,y}\varphi)(s,x) \Rightarrow (TC_{x,y}\varphi)(s,x) \quad \varphi(x,y) \Rightarrow \varphi(x,y)}{(TC_{x,y}\varphi)(s,x), \varphi(x,y) \Rightarrow (TC_{x,y}\varphi)(s,y)}\,(5.11)}{(TC_{x,y}\varphi)(s,r), (TC_{x,y}\varphi)(r,t) \Rightarrow (TC_{x,y}\varphi)(s,t)}\,(5.6)}{(TC_{x,y}\varphi)(r,t) \Rightarrow (TC_{x,y}\varphi)(s,t)}}{\Rightarrow (TC_{x,y}\varphi)(s,t)}$$

Note that in the application of Rule 5.6 we take $(TC_{x,y}\varphi)(s,x)$ to be ψ.

Thus, we find that TC_G and TC'_G are equivalent in the presence of the cut rule. So the reader might wonder why the more complicated version has been chosen. The answer to this lies within the proofs above. If we look at the above proofs we see that deriving 5.11 and 5.12 in TC_G does not involve the cut rule, whereas the above derivation of 5.5 in TC'_G does, and in the next section we show that the use of these cuts is actually unavoidable. Therefore, the rules will not be equivalent in any cut-free fragment. Choosing rule 5.5 increases the chance of achieving cut-elimination, a topic we will explore in section 5.5.

Since all the previous systems suggested for ancestral logic we know of (see [3, 4, 5]) worked with the reflexive form of the transitive closure operator, we also present a Gentzen-style system for it which is based on the the system RTC_H.

Definition 5.13 (*The system RTC_G*)**.**
The *system RTC_G* is defined by adding to the basic Gentzen's system for first-order logic $\mathcal{LK}_=$ the axiom:

$$\Gamma \Rightarrow \Delta, (RTC_{x,y}\varphi)(s,s) \tag{5.13}$$

and the following inference rules*:*

$$\frac{\Gamma \Rightarrow \Delta, \varphi\left\{\frac{s}{x}, \frac{t}{y}\right\}}{\Gamma \Rightarrow \Delta, (RTC_{x,y}\varphi)(s,t)} \tag{5.14}$$

$$\frac{\Gamma \Rightarrow \Delta, (RTC_{x,y}\varphi)(s,r) \quad \Gamma \Rightarrow \Delta, (RTC_{x,y}\varphi)(r,t)}{\Gamma \Rightarrow \Delta, (RTC_{x,y}\varphi)(s,t)} \tag{5.15}$$

$$\frac{\Gamma, \psi(x), \varphi(x,y) \Rightarrow \Delta, \psi\left\{\frac{y}{x}\right\}}{\Gamma, \psi\left\{\frac{s}{x}\right\}, (RTC_{x,y}\varphi)(s,t) \Rightarrow \Delta, \psi\left\{\frac{t}{x}\right\}} \tag{5.16}$$

The same restrictions on the rules in TC_G apply here.

By almost the same arguments, the same connection between the induction rule of PA_G and 5.16 can be established in RTC_G. Also, as in TC_G there are 2 variations of rule 5.17 which are equivalent:

$$\frac{\Gamma \Rightarrow \Delta, (RTC_{x,y}\varphi)(s,r) \quad \Gamma \Rightarrow \Delta, \varphi\left\{\frac{r}{x}, \frac{t}{y}\right\}}{\Gamma \Rightarrow \Delta, (RTC_{x,y}\varphi)(s,t)} \tag{5.17}$$

$$\frac{\Gamma \Rightarrow \Delta, \varphi\left\{\frac{s}{x}, \frac{r}{y}\right\} \quad \Gamma \Rightarrow \Delta, (RTC_{x,y}\varphi)(r,t)}{\Gamma \Rightarrow \Delta, (RTC_{x,y}\varphi)(s,t)} \tag{5.18}$$

Note that we could have replaced Rules 5.14 and 5.15 by one of the Rules 5.17 or 5.18. In this case Rules 5.14 and 5.15 will be derivable, but only by using cuts.

Since the two forms of the transitive closure operator can be expressed in terms of one another (see proposition 3.6), it is interesting to explore the connection between the two systems. Let φ be a sentence in $\mathcal{L}_{TC}$. φ^* is a sentence in $\mathcal{L}_{RTC}$ defined by induction as follows: for each sentence φ in first-order language define $\varphi^* := \varphi$, and define $((TC_{x,y}A)(s,t))^*$ to be the formula: $\exists z \left(A^*\left\{\frac{s}{x}, \frac{z}{y}\right\} \wedge (RTC_{x,y}A^*)(z,t)\right)$. Let ψ be a sentence in $\mathcal{L}_{RTC}$. Then ψ' is a sentence in $\mathcal{L}_{TC}$ defined by induction as follows: for each sentence ψ in first-order language define $\psi' := \psi$, and define $((RTC_{x,y}A)(s,t))'$ to be the formula: $(TC_{x,y}A')(s,t) \vee s = t$.

Lemma 5.14. *The following holds:*

- $\left(\varphi\left\{\frac{s}{x}, \frac{t}{y}\right\}\right)^* = \varphi^*\left\{\frac{s}{x}, \frac{t}{y}\right\}$ and $\left(\varphi\left\{\frac{s}{x}, \frac{t}{y}\right\}\right)' = \varphi'\left\{\frac{s}{x}, \frac{t}{y}\right\}$.
- $(\neg\varphi)^* = \neg\varphi^*$ *and* $(\neg\varphi)' = \neg\varphi'$.
- $(\varphi \circ \psi)^* = \varphi^* \circ \psi^*$ *and* $(\varphi \circ \psi)' = \varphi' \circ \psi'$, where $\circ \in \{\wedge, \vee, \rightarrow\}$.
- $(Qx\varphi)^* = Qx\varphi^*$ *and* $(Qx\varphi)' = Qx\varphi'$, where $Q \in \{\forall, \exists\}$.

Theorem 5.15. *Corresponding to any proof of* $\Gamma \Rightarrow \Delta$ *in* TC_G *there is a parallel proof in* RTC_G *of* $\Gamma^* \Rightarrow \Delta^*$*, and corresponding to any proof of* $\Gamma \Rightarrow \Delta$ *in* RTC_G *there is a parallel proof in* TC_G *of* $\Gamma' \Rightarrow \Delta'$*.*

Proof. We start by showing that for any proof of $\Gamma \Rightarrow \Delta$ in TC_G there is a parallel proof in RTC_G of $\Gamma^* \Rightarrow \Delta^*$. The proof is carried out by induction, we state here only the cases concerning the TC operator . In the following derivations we use Lemma 5.14, and again omit the context from the sequents in the derivations.

Case 1. Rule 5.4: An application of rule 5.4 can be transformed into the following derivation:

$$\dfrac{\dfrac{\Rightarrow \varphi^* \left\{\frac{s}{x}, \frac{t}{y}\right\} \qquad \Rightarrow \left(RTC_{x,y}\varphi^*\right)(t,t)}{\Rightarrow \varphi^* \left\{\frac{s}{x}, \frac{t}{y}\right\} \wedge \left(RTC_{x,y}\varphi^*\right)(t,t)}}{\Rightarrow \exists z \left(\varphi^* \left\{\frac{s}{x}, \frac{z}{y}\right\} \wedge \left(RTC_{x,y}\varphi^*\right)(z,t)\right)}$$

Case 2. Rule 5.5: We may use Rule 5.12, since the two rules are proven to be equivalent in TC_G. There is a proof in RTC_G of

$$\varphi^* \left\{\frac{r}{x}, \frac{z}{y}\right\} \wedge \left(RTC_{x,y}\varphi^*\right)(z,t) \Rightarrow \left(RTC_{x,y}\varphi^*\right)(r,t)$$

(using 5.18). Thus, there is a proof in RTC_G of

$$\exists z \left(\varphi^* \left\{\frac{r}{x}, \frac{z}{y}\right\} \wedge \left(RTC_{x,y}\varphi^*\right)(z,t)\right) \Rightarrow \left(RTC_{x,y}\varphi^*\right)(r,t)$$

Therefore, by the I.H., we get that from $\Rightarrow \exists z \left(\varphi^* \left\{\frac{r}{x}, \frac{z}{y}\right\} \wedge \left(RTC_{x,y}\varphi^*\right)(z,t)\right)$ and
$\Rightarrow \varphi^* \left\{\frac{s}{x}, \frac{r}{y}\right\}$ we can prove $\Rightarrow \exists z \left(\varphi^* \left\{\frac{s}{x}, \frac{z}{y}\right\} \wedge \left(RTC_{x,y}\varphi^*\right)(z,t)\right)$.

Case 3. Rule 5.6: An application of rule 5.6 can be transformed into the following derivation:

$$\dfrac{\dfrac{\dfrac{\psi^*(x), \varphi^*(x,y) \Rightarrow \psi^*\left\{\frac{y}{x}\right\}}{\psi^*\left\{\frac{s}{x}\right\}, \varphi^*\left\{\frac{s}{x}, \frac{z}{y}\right\} \Rightarrow \psi^*\left\{\frac{z}{x}\right\}}\,(sub) \qquad \dfrac{\psi^*(x), \varphi^*(x,y) \Rightarrow \psi^*\left\{\frac{y}{x}\right\}}{\psi^*\left\{\frac{z}{x}\right\}, \left(RTC_{x,y}\varphi^*\right)(z,t) \Rightarrow \psi^*\left\{\frac{t}{x}\right\}}\,(5.16)}{\psi^*\left\{\frac{s}{x}\right\}, \varphi^*\left\{\frac{s}{x}, \frac{z}{y}\right\} \wedge \left(RTC_{x,y}\varphi^*\right)(z,t) \Rightarrow \psi^*\left\{\frac{t}{x}\right\}}\,(cut + \wedge L)}{\psi^*\left\{\frac{s}{x}\right\}, \exists z \left(\varphi^*\left\{\frac{s}{x}, \frac{z}{y}\right\} \wedge \left(RTC_{x,y}\varphi^*\right)(z,t)\right) \Rightarrow \psi^*\left\{\frac{t}{x}\right\}}$$

Next we show that for any proof of $\Gamma \Rightarrow \Delta$ in RTC_G there is a parallel proof in TC_G of $\Gamma' \Rightarrow \Delta'$. This too is done by induction, and we only present here the cases concerning the RTC operator. In the following derivations we use Lemma 5.14.

Case 1. Axiom 5.13: The immediate translation of this axiom is $\Gamma' \Rightarrow \Delta', (TC_{x,y}\varphi')(s,s) \vee s = s$, which is easily derivable from an axiom of $\mathcal{LK}_=$.

Case 2. Rule 5.14: An application of rule 5.14 can be transformed into the following derivation (using Remark 5.10):

$$\frac{\Rightarrow \varphi' \left\{\frac{s}{x}, \frac{t}{y}\right\}}{\Rightarrow (TC_{x,y}\varphi')(s,t), s = t} \ (5.4 + wkR)$$

Case 3. Rule 5.15: An application of rule 5.15 can be transformed into the following derivation (using Remark 5.10):

$$\frac{\Rightarrow (TC_{x,y}\varphi')(s,r), s = r, \quad \Rightarrow (TC_{x,y}\varphi')(r,t), r = t}{\Rightarrow (TC_{x,y}\varphi')(s,t), s = t} \ (5.5 + eq)$$

Case 4. Rule 5.16: An application of rule 5.16 can be transformed into the following derivation:

$$\frac{\dfrac{\psi'(x), \varphi'(x,y) \Rightarrow \psi'\left\{\frac{y}{x}\right\}}{\psi'\left\{\frac{s}{x}\right\}, (TC_{x,y}\varphi')(s,t) \Rightarrow \psi'\left\{\frac{t}{x}\right\}} \ (5.6) \qquad \psi'\left\{\frac{s}{x}\right\}, s = t \Rightarrow \psi'\left\{\frac{t}{x}\right\}}{\psi'\left\{\frac{s}{x}\right\}, (TC_{x,y}\varphi')(s,t) \vee s = t \Rightarrow \psi'\left\{\frac{t}{x}\right\}}$$

□

The last theorem entails that any theorem of TC_G can be translated to a theorem in RTC_G and vice versa. However, the more interesting result is that all fundamental rules concerning RTC that have been suggested (as far as we know) are derivable in RTC_G, while in TC_G this is not the case. i.e. there are fundamental rules regarding the TC operator which are not derivable in TC_G.

Proposition 5.16. *The following rules are derivable in RTC_G:*[10]

$$\frac{\Gamma \Rightarrow \Delta, \left(RTC_{x,y}\varphi\right)(s,t)}{\Gamma \Rightarrow \Delta, s=t, \exists z\left(\left(RTC_{x,y}\varphi\right)(s,z) \wedge \varphi\left\{\frac{z}{x},\frac{t}{y}\right\}\right)} \qquad \frac{\Gamma \Rightarrow \Delta, \left(RTC_{x,y}\varphi\right)(s,t)}{\Gamma \Rightarrow \Delta, s=t, \exists z\left(\varphi\left\{\frac{s}{x},\frac{z}{y}\right\} \wedge \left(RTC_{x,y}\varphi\right)(z,t)\right)} \tag{5.19}$$

$$\frac{\Gamma \Rightarrow \Delta, \left(RTC_{x,y}\varphi\right)(s,t)}{\Gamma \Rightarrow \Delta, \left(RTC_{y,x}\varphi\right)(t,s)} \qquad \frac{\left(RTC_{x,y}\varphi\right)(s,t), \Gamma \Rightarrow \Delta}{\left(RTC_{y,x}\varphi\right)(t,s), \Gamma \Rightarrow \Delta} \tag{5.20}$$

$$\frac{\Gamma \Rightarrow \Delta, \left(RTC_{x,y}\varphi\right)(s,t)}{\Gamma \Rightarrow \Delta, \left(RTC_{u,v}\varphi\left\{\frac{u}{x},\frac{v}{y}\right\}\right)(s,t)} \qquad \frac{\left(RTC_{x,y}\varphi\right)(s,t), \Gamma \Rightarrow \Delta}{\left(RTC_{u,v}\varphi\left\{\frac{u}{x},\frac{v}{y}\right\}\right)(s,t), \Gamma \Rightarrow \Delta} \tag{5.21}$$

$$\frac{\Gamma, \varphi \Rightarrow \Delta, \psi}{\Gamma, \left(RTC_{x,y}\varphi\right)(s,t) \Rightarrow \Delta, \left(RTC_{x,y}\psi\right)(s,t)} \tag{5.22}$$

$$\frac{\left(RTC_{x,y}\varphi\right)(s,t), \Gamma \Rightarrow \Delta}{\left(RTC_{u,v}\left(RTC_{x,y}\varphi\right)(u,v)\right)(s,t), \Gamma \Rightarrow \Delta} \tag{5.23}$$

$$\frac{\varphi\left\{\frac{s}{x}\right\}, \Gamma \Rightarrow \Delta}{\left(RTC_{x,y}\varphi\right)(s,t), \Gamma \Rightarrow s=t, \Delta} \qquad \frac{\varphi\left\{\frac{t}{y}\right\}, \Gamma \Rightarrow \Delta}{\left(RTC_{x,y}\varphi\right)(s,t), \Gamma \Rightarrow s=t, \Delta} \tag{5.24}$$

Conditions:

- In all the rules we assume that the terms which are substituted are free for substitution and that no forbidden capturing occurs.
- In 5.19 z should not occur free in Γ, Δ and $\varphi\left\{\frac{s}{x},\frac{t}{y}\right\}$.
- In 5.21 the conditions are the usual ones concerning the α-rule.
- In 5.22 x, y should not occur free in Γ, Δ.
- In 5.23 u, v should not occur free in φ.
- In 5.24 y should not occur free in Γ, Δ or s in the left rule, and x should not occur free in Γ, Δ or t in the right rule.

[10] These rules were suggested in [1, 3, 4, 5].

Proof. In all the derivations in this proof we use freely Remark 5.10, and again omit the context Γ, Δ from the sequents in the derivations.

- The first rule in 5.19: Consider the following proofs:
  P_1 :

$$\dfrac{\dfrac{\dfrac{\Rightarrow (RTC_{x,y}\varphi)\,(y,y)}{s=y \Rightarrow (RTC_{x,y}\varphi)\,(s,y)}\,(eq3) \qquad \varphi\left\{\frac{y}{x},\frac{z}{y}\right\} \Rightarrow \varphi\left\{\frac{y}{x},\frac{z}{y}\right\}}{s=y, \varphi\left\{\frac{y}{x},\frac{z}{y}\right\} \Rightarrow (RTC_{x,y}\varphi)\,(s,y) \wedge \varphi\left\{\frac{y}{x},\frac{z}{y}\right\}}}{s=y, \varphi\left\{\frac{y}{x},\frac{z}{y}\right\} \Rightarrow \exists w \left((RTC_{x,y}\varphi)\,(s,w) \wedge \varphi\left\{\frac{w}{x},\frac{z}{y}\right\}\right)}$$

  The sequent $(RTC_{x,y}\varphi)\,(s,w), \varphi\left\{\frac{w}{x},\frac{y}{y}\right\} \Rightarrow (RTC_{x,y}\varphi)\,(s,y)$ is provable in RTC_G. Thus, we can construct the following P_2 :

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\vdots\\ (RTC_{x,y}\varphi)\,(s,w) \wedge \varphi\left\{\frac{w}{x},\frac{y}{y}\right\} \Rightarrow (RTC_{x,y}\varphi)\,(s,y)\end{matrix}}{\exists w \left((RTC_{x,y}\varphi)\,(s,w) \wedge \varphi\left\{\frac{w}{x},\frac{y}{y}\right\}\right) \Rightarrow (RTC_{x,y}\varphi)\,(s,y)} \quad \varphi\left\{\frac{y}{x},\frac{z}{y}\right\} \Rightarrow \varphi\left\{\frac{y}{x},\frac{z}{y}\right\}}{\exists w \left((RTC_{x,y}\varphi)\,(s,w) \wedge \varphi\left\{\frac{w}{x},\frac{y}{y}\right\}\right), \varphi\left\{\frac{y}{x},\frac{z}{y}\right\} \Rightarrow (RTC_{x,y}\varphi)\,(s,y) \wedge \varphi\left\{\frac{y}{x},\frac{z}{y}\right\}}}{\exists w \left((RTC_{x,y}\varphi)\,(s,w) \wedge \varphi\left\{\frac{w}{x},\frac{y}{y}\right\}\right), \varphi\left\{\frac{y}{x},\frac{z}{y}\right\} \Rightarrow \exists w \left((RTC_{x,y}\varphi)\,(s,w) \wedge \varphi\left\{\frac{w}{x},\frac{z}{y}\right\}\right)}$$

  From P_1 and P_2 we can obtain:

$$\dfrac{\dfrac{\dfrac{\vdots\, P_1 \qquad \vdots\, P_2}{\exists w \left((RTC_{x,y}\varphi)\,(s,w) \wedge \varphi\left\{\frac{w}{x},\frac{y}{y}\right\}\right) \vee s=y, \varphi\left\{\frac{y}{x},\frac{z}{y}\right\} \Rightarrow \exists w \left((RTC_{x,y}\varphi)\,(s,w) \wedge \varphi\left\{\frac{w}{x},\frac{z}{y}\right\}\right) \vee s=z}}{\exists w \left((RTC_{x,y}\varphi)\,(s,w) \wedge \varphi\left\{\frac{w}{x},\frac{s}{y}\right\}\right) \vee s=s, (RTC_{x,y}\varphi)\,(s,t) \Rightarrow \exists w \left((RTC_{x,y}\varphi)\,(s,w) \wedge \varphi\left\{\frac{w}{x},\frac{t}{y}\right\}\right) \vee s=t}\,(5.16)}{(RTC_{x,y}\varphi)\,(s,t) \Rightarrow \exists w \left((RTC_{x,y}\varphi)\,(s,w) \wedge \varphi\left\{\frac{w}{x},\frac{t}{y}\right\}\right) \vee s=t}\,(eq+cut)$$

The proof for the second rule in 5.19 is symmetric.

- The left rule in 5.20: Consider the following proof P_1, where P' is obtained using Rule 5.19:

$$\dfrac{\begin{array}{c}\vdots\ P'\\ (RTC_{x,y}\varphi)(s,t)\Rightarrow s=t,\exists z\left(\varphi\left\{\frac{s}{x},\frac{z}{y}\right\}\wedge(RTC_{x,y}\varphi)(z,t)\right)\end{array}\qquad\dfrac{\Rightarrow(RTC_{y,x}\varphi)(s,s)}{s=t\Rightarrow(RTC_{y,x}\varphi)(t,s)}}{(RTC_{x,y}\varphi)(s,t)\Rightarrow(RTC_{y,x}\varphi)(t,s),\exists z\left(\varphi\left\{\frac{s}{x},\frac{z}{y}\right\}\wedge(RTC_{x,y}\varphi)(z,t)\right)}$$

The sequent $\varphi(x,y),(RTC_{y,x}\varphi)(x,s)\Rightarrow(RTC_{y,x}\varphi)(y,s)$ is provable in RTC_G using 5.18. Thus, we can construct the following P_2 :

$$\dfrac{\dfrac{\dfrac{\varphi\left\{\frac{z}{y},\frac{s}{x}\right\}\Rightarrow\varphi\left\{\frac{z}{y},\frac{s}{x}\right\}\quad\Rightarrow(RTC_{y,x}\varphi)(s,s)}{\varphi\left\{\frac{z}{y},\frac{s}{x}\right\}\Rightarrow(RTC_{y,x}\varphi)(z,s)}\,(5.18)\quad\dfrac{\varphi(x,y),(RTC_{y,x}\varphi)(x,s)\Rightarrow(RTC_{y,x}\varphi)(y,s)}{(RTC_{x,y}\varphi)(z,t),(RTC_{y,x}\varphi)(z,s)\Rightarrow(RTC_{y,x}\varphi)(t,s)}\,(5.16)}{\varphi\left\{\frac{s}{x},\frac{z}{y}\right\}\wedge(RTC_{x,y}\varphi)(z,t)\Rightarrow(RTC_{y,x}\varphi)(t,s)}}{\exists z\left(\varphi\left\{\frac{s}{x},\frac{z}{y}\right\}\wedge(RTC_{x,y}\varphi)(z,t)\right)\Rightarrow(RTC_{y,x}\varphi)(t,s)}$$

Using a cut, from P_1 and P_2 we obtain:

$$\vdash_{RTC_G}(RTC_{x,y}\varphi)(s,t)\Rightarrow(RTC_{y,x}\varphi)(t,s)$$

The proof of the right rule is symmetric.

- The left rule in 5.21: We have that

$$\vdash_{RTC_G} s=t\Rightarrow\left(RTC_{u,v}\varphi\left\{\frac{u}{x},\frac{v}{y}\right\}\right)(s,t)$$

By a similar method to the one used in the proof of 5.20 we get:

$$\vdash_{RTC_G}\exists z\left((RTC_{x,y}\varphi)(s,z)\wedge\varphi\left\{\frac{z}{x},\frac{t}{y}\right\}\right)\Rightarrow\left(RTC_{u,v}\varphi\left\{\frac{u}{x},\frac{v}{y}\right\}\right)(s,t)$$

Thus using cuts and Rule 5.19 we obtain:

$$\vdash_{RTC_G}(RTC_{x,y}\varphi)(s,t)\Rightarrow\left(RTC_{u,v}\varphi\left\{\frac{u}{x},\frac{v}{y}\right\}\right)(s,t)$$

The proof of the right rule is symmetric.

- Rule 5.22: Consider the following proof P_1, where P' is obtained using Rule 5.19:

$$\dfrac{\begin{array}{c}\vdots\ P'\\ (RTC_{x,y}\varphi)(s,t)\Rightarrow s=t,\exists z\left(\varphi\left\{\frac{s}{x},\frac{z}{y}\right\}\wedge(RTC_{x,y}\varphi)(z,t)\right)\end{array}\qquad\dfrac{\Rightarrow RTC_{x,y}\psi(s,s)}{s=t\Rightarrow RTC_{x,y}\psi(s,t)}}{(RTC_{x,y}\varphi)(s,t)\Rightarrow(RTC_{x,y}\psi)(s,t),\exists z\left(\varphi\left\{\frac{s}{x},\frac{z}{y}\right\}\wedge(RTC_{x,y}\varphi)(z,t)\right)}\,(cut)$$

Then, observe the following proof P_2:

$$\dfrac{\dfrac{\dfrac{\dfrac{\varphi \Rightarrow \psi}{\varphi\left\{\frac{s}{x},\frac{z}{y}\right\} \Rightarrow \psi\left\{\frac{s}{x},\frac{z}{y}\right\}}\,(sub) \qquad \Rightarrow \left(RTC_{x,y}\psi\right)(z,z)}{\varphi\left\{\frac{s}{x},\frac{z}{y}\right\} \Rightarrow \left(RTC_{x,y}\psi\right)(s,z)}\,(5.18) \qquad \dfrac{\dfrac{\left(RTC_{x,y}\psi\right)(s,z) \Rightarrow \left(RTC_{x,y}\psi\right)(s,z) \quad \dfrac{\varphi \Rightarrow \psi}{\varphi\left\{\frac{z}{x},\frac{u}{y}\right\} \Rightarrow \psi\left\{\frac{z}{x},\frac{u}{y}\right\}}}{\left(RTC_{x,y}\psi\right)(s,z), \varphi\left\{\frac{z}{x},\frac{u}{y}\right\} \Rightarrow \left(RTC_{x,y}\psi\right)(s,u)}\,(5.17)}{\left(RTC_{x,y}\psi\right)(s,z), \left(RTC_{x,y}\varphi\right)(z,t) \Rightarrow \left(RTC_{x,y}\psi\right)(s,t)}\,(5.16)}{\varphi\left\{\frac{s}{x},\frac{z}{y}\right\} \wedge \left(RTC_{x,y}\varphi\right)(z,t) \Rightarrow \left(RTC_{x,y}\psi\right)(s,t)}\,(cut)}{\exists z\left(\varphi\left\{\frac{s}{x},\frac{z}{y}\right\} \wedge \left(RTC_{x,y}\varphi\right)(z,t)\right) \Rightarrow \left(RTC_{x,y}\psi\right)(s,t)}$$

From P_1 and P_2 we can obtain: $\vdash_{RTC_G} \left(RTC_{x,y}\varphi\right)(s,t) \Rightarrow \left(RTC_{x,y}\psi\right)(s,t)$.[11]

- Rule 5.23: Note that

  $$\vdash_{RTC_G} \left(RTC_{x,y}\varphi\right)(s,u), \left(RTC_{x,y}\varphi\right)(u,v) \Rightarrow \left(RTC_{x,y}\varphi\right)(s,v)$$

  Then, by applying Rule 5.16 we obtain:

  $$\vdash_{RTC_G} \left(RTC_{x,y}\varphi\right)(s,s), \left(RTC_{u,v}\left(RTC_{x,y}\varphi\right)(u,v)\right)(s,t) \Rightarrow \left(RTC_{x,y}\varphi\right)(s,t)$$

  Since $\Rightarrow \left(RTC_{x,y}\varphi\right)(s,s)$ is an axiom, using a cut we can obtain

  $$\vdash_{RTC_G} \left(RTC_{u,v}\left(RTC_{x,y}\varphi\right)(u,v)\right)(s,t) \Rightarrow \left(RTC_{x,y}\varphi\right)(s,t)$$

- The left rule in 5.24: Note that by the second rule in 5.19 we can obtain $\vdash_{RTC_G} \left(RTC_{x,y}\varphi\right)(s,t) \Rightarrow s = t, \exists z\left(\varphi\left\{\frac{s}{x},\frac{z}{y}\right\} \wedge \left(RTC_{x,y}\varphi\right)(z,t)\right)$. From the sequent $\varphi\left\{\frac{s}{x}\right\} \Rightarrow$ we can derive by standard rules of $\mathcal{LK}_=$ the sequent: $\exists z\left(\varphi\left\{\frac{s}{x},\frac{z}{y}\right\} \wedge \left(RTC_{x,y}\varphi\right)(z,t)\right) \Rightarrow$, where z is a fresh variable. Thus, by using the cut rule we can obtain: $\vdash_{RTC_G} \left(RTC_{x,y}\varphi\right)(s,t) \Rightarrow s = t$. The proof of the right rule in 5.24 is similar, only this time we use the first rule in 5.19.

□

The last theorem provides a partial evidence for the claim that RTC_G is the appropriate system for ancestral logic, since it can derive all fundamental rules concerning the RTC operator that have been suggested in the literature. Unfortunately, this is not the case in TC_G. There are fundamental properties of the TC operator which cannot be derived in TC_G.

[11] Note that the cut in P_1 is inessential, while the one in P_2 is essential.

Theorem 5.17. *The following valid rules are not derivable in TC_G:*

$$\frac{\varphi\left\{\frac{s}{x}\right\},\Gamma\Rightarrow\Delta}{(TC_{x,y}\varphi)(s,t),\Gamma\Rightarrow\Delta}\qquad\frac{\varphi\left\{\frac{t}{y}\right\},\Gamma\Rightarrow\Delta}{(TC_{x,y}\varphi)(s,t),\Gamma\Rightarrow\Delta}\tag{5.25}$$

$$\begin{gathered}\frac{\Gamma\Rightarrow\Delta,(TC_{x,y}\varphi)(s,t)}{\Gamma\Rightarrow\Delta,\varphi\left\{\frac{s}{x},\frac{t}{y}\right\},\exists z\left((TC_{x,y}\varphi)(s,z)\wedge\varphi\left\{\frac{z}{x},\frac{t}{y}\right\}\right)}\\ \frac{\Gamma\Rightarrow\Delta,(TC_{x,y}\varphi)(s,t)}{\Gamma\Rightarrow\Delta,\varphi\left\{\frac{s}{x},\frac{t}{y}\right\},\exists z\left(\varphi\left\{\frac{s}{x},\frac{z}{y}\right\}\wedge(TC_{x,y}\varphi)(z,t)\right)}\end{gathered}\tag{5.26}$$

where in 5.25 y should not occur free in Γ,Δ or s in the left rule, and x should not occur free in Γ,Δ or t in the right rule; and in 5.26 z should not occur free in Γ,Δ and $\varphi\left\{\frac{s}{x},\frac{t}{y}\right\}$.

Proof. It is easy to see that all the rules of TC_G remain valid and derivable in RTC_G if we replace the operator TC with RTC. Assume that the above rules are derivable in TC_G. Hence, we would get that in RTC_G the following are provable:

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

$$\begin{gathered}\frac{\Gamma\Rightarrow\Delta,(RTC_{x,y}\varphi)(s,t)}{\Gamma\Rightarrow\Delta,\varphi\left\{\frac{s}{x},\frac{t}{y}\right\},\exists z\left((RTC_{x,y}\varphi)(s,z)\wedge\varphi\left\{\frac{z}{x},\frac{t}{y}\right\}\right)}\\ \frac{\Gamma\Rightarrow\Delta,(RTC_{x,y}\varphi)(s,t)}{\Gamma\Rightarrow\Delta,\varphi\left\{\frac{s}{x},\frac{t}{y}\right\},\exists z\left(\varphi\left\{\frac{s}{x},\frac{z}{y}\right\}\wedge(RTC_{x,y}\varphi)(z,t)\right)}\end{gathered}$$

Which are obviously not valid rules of RTC_G, since $(RTC_{x,y}\varphi)(s,s)$ holds for all s and φ. □

In general, any rule which is valid only for the TC operator and not to the RTC operator will not be derivable in TC_G. So, the question that arises from the last theorem is what do we need to add to the system TC_G in order to make it able to derive all the basic theorems rules the TC operator.

Proposition 5.18. *If we add to TC_G the axiom:*

$$(TC_{x,y}\varphi)(s,t) \Rightarrow \varphi\left\{\frac{s}{x},\frac{t}{y}\right\}, \exists z\left((TC_{x,y}\varphi)(s,z) \wedge \varphi(z,t)\right) \tag{5.27}$$

or its equivalent:

$$(TC_{x,y}\varphi)(s,t) \Rightarrow \varphi\left\{\frac{s}{x},\frac{t}{y}\right\}, \exists w\left(\varphi(s,z) \wedge (TC_{x,y}\varphi)(z,t)\right) \tag{5.28}$$

where z does not occur free in Γ, Δ and $\varphi\left\{\frac{s}{x},\frac{t}{y}\right\}$, we get a system strong enough to derive all the TC-counterparts of the rules in Proposition 5.16.

The proof of the last theorem is given by derivations similar to those in Proposition 5.16. In each of the proofs we replace the use of one of the rules in 5.19 by the corresponding axiom for the TC operator (Axiom 5.27 or 5.28).

Axioms 5.27 and 5.28 are obviously too complicated to be taken as axioms if we want the system to remain effective. Further research is required in order to find what other "simple" rules or axioms can be added to TC_G in order to make these axioms derivable.

5.4 The connection to PA

From the last discussion we get that from the two systems for ancestral logic, the system RTC_G is the better candidate to be used as a proof system for ancestral logic. One support for this claim comes from the following connection with PA_G.

Definition 5.19. The system RTC_A is obtained from RTC_G by adding the following axioms:

$$\begin{gathered} s(x) = 0 \Rightarrow \\ s(x) = s(y) \Rightarrow x = y \\ \Rightarrow x + 0 = x \\ \Rightarrow x + s(y) = s(x+y) \\ \Rightarrow (RTC_{w,u}(s(w) = u))(0, x) \end{gathered}$$

Note. Using the definition for multiplication given in Theorem 3.14(1), replacing in it formulas of the form $(TC_{x,y}A)\,(s,t)$ by $\exists z \left(A\left\{\frac{s}{x},\frac{z}{y}\right\} \wedge (RTC_{x,y}A)\,(z,t)\right)$, we get that the axioms for multiplication (2.5 and 2.6) are derivable in RTC_A. The proof is straightforward and long so we omit it.

Theorem 5.20. *The system RTC_A is equivalent to PA_G.*

Proof. Due to the former discussion on the connection between the induction rules of the systems, we get that RTC_A is an extension of PA_G. Thus, RTC_A can prove any theorem of the original PA_G. For the converse, define a translation of each sentence φ in the language of RTC for arithmetics, to a sentence φ^* in the language of PA. Then we prove that if $\vdash_{RTC_A} \Gamma \Rightarrow \Delta$, then $\vdash_{PA_G} \Gamma^* \Rightarrow \Delta^*$. For the translation we use a beta function which allows us to encode in PA finite sequences (this idea is taken from [2]). 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 $i \leq n$, $\beta(c,i) = k_i$.Thus, our motivation is that is s,t are closed terms, $(TC_{x,y}\varphi)\,(s,t)$ holds iff for some n, there is a sequence $k_0, k_1, ..., k_n$ such that $k_0 = I\,[s]$, $k_n = I\,[t]$, and each pair of consecutive terms are in the relation defined by φ. Using a two place β-function, $(RTC_{x,y}\varphi)\,(s,t)$ is true in $\mathcal{N}$ iff there are n, c such that the following hold:

- $\beta(c,0) = I\,[s]$
- $\beta(c,n) = I\,[t]$
- $\forall i < n \quad \varphi\left\{\frac{\beta(c,i)}{x}, \frac{\beta(c,s(i))}{y}\right\} \vee (\beta(c,i) = \beta(c,s(i)))$ is true in $\mathcal{N}$.

Accordingly, let B be a wff of $\mathcal{L}_{PA}$ with three free variables which captures in PA a β-function[12]. For each sentence φ in first-order language define $\varphi^* := \varphi$. Define $((RTC_{x,y}\varphi)\,(s,t))^*$ to be the following formula:

$$\exists z \exists c \left(B\,(c,0,s) \wedge B\,(c,z,t) \wedge \forall_{u<z} \exists v \exists w \left(B\,(c,u,v) \wedge B\,(c,s\,(u)\,,w) \wedge \left(\varphi * \left\{\frac{v}{x}, \frac{w}{y}\right\} \vee v = w\right)\right)\right)$$

It is easy to check that all the inference rules for the RTC-operator apply equally to starred wwf in PA_G, and the starred analogue of the axiom:

$$\Rightarrow (RTC_{w,u}\,(s(w) = u))\,(0,x)$$

[12] A theory T captures a two-place function f by a wff $\psi(x,y,z)$ iff, for any closed terms a,b,c: if $f\,(I\,[a]\,,I\,[b]) = I\,[c]$, then $T \vdash \psi(a,b,c)$; and if $f\,(I\,[a]\,,I\,[b]) \neq I\,[c]$, then $T \vdash \neg\psi(a,b,c)$.

is also a theorem of PA_G. So corresponding to any proof of $\Gamma \Rightarrow \Delta$ in RTC_A there is a parallel proof in plain PA_G of $\Gamma^* \Rightarrow \Delta^*$.

□

Experience shows that PA_G is a very natural and robust system for arithmetic. Despite efforts to find other examples, all arithmetical statements which are known not to be provable in PA_G are connected with Godel's incompleteness theorem. This provides an important evidence for the claim that RTC_A is a natural system for dealing with the formalization of mathematics.

Let us conclude. If we add to the basic first-order system for arithmetic the ancestral operator (in its reflexive form) we basically form a logical system which incorporate new ideas that do *not* go beyond those needed for us in order to understand elementary arithmetic and logic. Thus, we may claim that if one comes up with proofs of theorems which are unsettled by RTC_G, such proofs will have to go beyond the understanding of the ancestral. This entails a thesis which is our version of Isaacson's thesis[13] [17]:

Thesis. *If there is a proof of any true sentence of $\mathcal{L}_{RTC}$ which is independent of RTC_A, then we need for it ideas that go beyond those needed in order to understand $\mathcal{L}_{RTC}$.*

5.5 On cut elimination and constructive consistency proofs

Next we examine some fundamental proof-theoretic properties of TC_G and RTC_G, the most important of which is cut elimination. The cut elimination theorem states that the cut rule is admissible. Since our systems for ancestral logic include equality, we use an alternative version of the cut elimination theorem. A cut is said to be inessential if the cut formula is of the form $s = t$, otherwise it is called essential cut. A system with equality is said to admit cut-elimination if all essential cuts are admissible. In this section we shall give an overview of possible methods for syntactically proving that TC_G admits cut-elimination. In the discussion in this section about the cut elimination theorem we will refer to TC_G, though the same considerations show that all of the results apply to RTC_G as well.

[13] Since by Theorem 5.20 we know that RTC_A is equivalent to PA_G, Isaacson's thesis and this thesis are equivalent.

In the semantical proofs of cut elimination one usually establishes not only closure under cut, but also completeness. However, this type of proof does not provide a constructive method for eliminating cuts from a given proof. Syntactically eliminating essential cuts from a proof is not simply a matter of showing that the cut rule remains admissible if it is deleted from the list of the rules of the system. One should provide an algorithm for transforming any proof containing essential cuts to an essential-cut-free proof. The syntactic cut elimination proofs (see [10, 13, 8, 9]) use a method of going over a given proof and "reducing" it to a less complicated proof in some sense, until all essential cuts are eliminated. What is reduced can be the complexity of the cut formula, the "depth" of the proof, the ordinal of the proof or some other measure for the complexity of the proof. For example, Gentzen's classic proof of the cut-elimination theorem for first-order logic [8] uses a double induction: the main induction is on the number of logical connectives and quantifiers in the cut formula, and the sub-induction is on the "rank" of the cut, which is some measure depending on the place of the cut in the proof . A reduction step is defined for every derivation ending with an application of the cut rule. For instance, a cut on a compound formula is replaced by cuts on its sub-formulas, which necessarily contain a smaller number of connectives. Let us demonstrate. The derivation

$$\dfrac{\dfrac{\begin{array}{c}\vdots\, P_{11}\\ \Gamma \Rightarrow \Delta, \varphi\end{array} \quad \begin{array}{c}\vdots\, P_{12}\\ \Gamma \Rightarrow \Delta, \psi\end{array}}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi}\,(\wedge R) \quad \dfrac{\begin{array}{c}\vdots\, P_{21}\\ \varphi, \Gamma \Rightarrow \Delta\end{array}}{\varphi \wedge \psi, \Gamma \Rightarrow \Delta}\,(\wedge L)}{\Gamma \Rightarrow \Delta}$$

is reduced to

$$\dfrac{\begin{array}{c}\vdots\, P_{11}\\ \Gamma \Rightarrow \Delta, \varphi\end{array} \quad \begin{array}{c}\vdots\, P_{21}\\ \varphi, \Gamma \Rightarrow \Delta\end{array}}{\Gamma \Rightarrow \Delta}$$

By the induction hypothesis, this cut on φ can be eliminated, hence the original cut on $\varphi \wedge \psi$ can also be eliminated.

In TC_G things are much more complicated. Observe the following derivation in which a TC-formula is introduced on the right using Rule 5.4 and on the left using Rule 5.6, and then a cut is made on the TC-formula. Again, we shall omit the context from the sequents in the following derivations.

$$\dfrac{\dfrac{\begin{matrix}\vdots\, P_{11}\\ \Rightarrow \varphi\left\{\frac{s}{x},\frac{t}{y}\right\}\end{matrix}}{\Rightarrow (TC_{x,y}\varphi)\,(s,t)}\,(5.4) \quad \dfrac{\begin{matrix}\vdots\, P_{21}\\ \psi_{(x)},\varphi_{(x,y)} \Rightarrow \psi\left\{\frac{y}{x}\right\}\end{matrix}}{\psi\left\{\frac{s}{x}\right\},(TC_{x,y}\varphi)\,(s,t) \Rightarrow \psi\left\{\frac{t}{x}\right\}}\,(5.6)}{\psi\left\{\frac{s}{x}\right\} \Rightarrow \psi\left\{\frac{t}{x}\right\}}$$

The natural reduction of this part of the proof is:

$$\dfrac{\begin{matrix}\vdots\, P_{11}\\ \Rightarrow \varphi\left\{\frac{s}{x},\frac{t}{y}\right\}\end{matrix} \quad \dfrac{\begin{matrix}\vdots\, P_{21}\\ \psi_{(x)},\varphi_{(x,y)} \Rightarrow \psi\left\{\frac{y}{x}\right\}\end{matrix}}{\psi\left\{\frac{s}{x}\right\},\varphi\left\{\frac{s}{x},\frac{t}{y}\right\} \Rightarrow \psi\left\{\frac{t}{x}\right\}}}{\psi\left\{\frac{s}{x}\right\} \Rightarrow \psi\left\{\frac{t}{x}\right\}}$$

The cut on the formula $(TC_{x,y}\varphi)\,(s,t)$ is replaced by a cut on the formula $\varphi\left\{\frac{s}{x},\frac{t}{y}\right\}$ which is of smaller complexity. Hence, in this case we have a natural reduction in the proof.

However, let us examine the case in which a TC-formula is introduced on the right using Rule 5.5 and on the left using Rule 5.6, and then a cut is made on the TC-formula.

$$\dfrac{\dfrac{\begin{matrix}\vdots\, P_{11}\\ \Rightarrow (TC_{x,y}\varphi)\,(s,r)\end{matrix} \quad \begin{matrix}\vdots\, P_{12}\\ \Rightarrow (TC_{x,y}\varphi)\,(r,t)\end{matrix}}{\Rightarrow (TC_{x,y}\varphi)\,(s,t)}\,(5.5) \quad \dfrac{\begin{matrix}\vdots\, P_{21}\\ \psi_{(x)},\varphi_{(x,y)} \Rightarrow \psi\left\{\frac{y}{x}\right\}\end{matrix}}{\psi\left\{\frac{s}{x}\right\},(TC_{x,y}\varphi)\,(s,t) \Rightarrow \psi\left\{\frac{t}{x}\right\}}\,(5.6)}{\psi\left\{\frac{s}{x}\right\} \Rightarrow \psi\left\{\frac{t}{x}\right\}}$$

The natural reduction for part of the proof is:

$$\dfrac{\dfrac{\begin{matrix}\vdots\, P_{11}\\ \Rightarrow (TC_{x,y}\varphi)\,(s,r)\end{matrix} \quad \dfrac{\begin{matrix}\vdots\, P_{21}\\ \psi_{(x)},\varphi_{(x,y)} \Rightarrow \psi\left\{\frac{y}{x}\right\}\end{matrix}}{\psi\left\{\frac{s}{x}\right\},(TC_{x,y}\varphi)\,(s,r) \Rightarrow \psi\left\{\frac{r}{x}\right\}}\,(5.6)}{\psi\left\{\frac{s}{x}\right\} \Rightarrow \psi\left\{\frac{r}{x}\right\}} \quad \dfrac{\begin{matrix}\vdots\, P_{12}\\ \Rightarrow (TC_{x,y}\varphi)\,(r,t)\end{matrix} \quad \dfrac{\begin{matrix}\vdots\, P_{21}\\ \psi_{(x)},\varphi_{(x,y)} \Rightarrow \psi\left\{\frac{y}{x}\right\}\end{matrix}}{\psi\left\{\frac{r}{x}\right\},(TC_{x,y}\varphi)\,(r,t) \Rightarrow \psi\left\{\frac{t}{x}\right\}}\,(5.6)}{\psi\left\{\frac{s}{x}\right\} \Rightarrow \psi\left\{\frac{r}{x}\right\}}}{\psi\left\{\frac{s}{x}\right\} \Rightarrow \psi\left\{\frac{t}{x}\right\}}$$

Here, the cut on the formula $(TC_{x,y}\varphi)(s,t)$ is replaced by three cuts on the formulas: $(TC_{x,y}\varphi)(r,t)$, $(TC_{x,y}\varphi)(s,r)$ and $\psi\left\{\frac{r}{x}\right\}$. It is unclear what kind of measure can be used here in order to achieve a reduction in the proof. The number of applications of Rule 5.5 has gone down by one, yet the duplication of the derivation ending with $\psi_{(x)}, \varphi_{(x,y)} \Rightarrow \psi\left\{\frac{y}{x}\right\}$ and the application of the induction rule might offset this. Another, much more crucial difficulty is the following: while the two new cut formulas, $(TC_{x,y}\varphi)(r,t)$ and $(TC_{x,y}\varphi)(s,r)$, are of complexity equal to that of the original cut formula $(TC_{x,y}\varphi)(s,t)$ and there is reduction of the depth, the real difficulty is that the new cut formula $\psi\left\{\frac{r}{x}\right\}$ is not related at all to the original cut formula. Thus it can be of larger complexity than $(TC_{x,y}\varphi)(s,t)$, unless we force some constraints on the applicability of the induction rule.

While it might be possible to overcome these difficulties using some restrictions, there is still a much more fundamental problem connected with the induction rule. So far we only examined cases in which the cut is made on the principal formula of the premises (in our case - a TC-formula), but what about other cases? For instance, it is unclear what should be the reductions in cases where one of the premises is the conclusion of the induction rule and the cut is made on the induction formula. A good example for such a case is the proof of 5.5 in TC'_G. It is easy to see that we cannot eliminate the cuts using the natural reductions just described since doing so prevents us from applying the induction rule.

In order to avoid this problem, Gentzen applied a different method for the induction rule when proving the consistency of PA_G [8, 9]. The consistency proof of PA_G is based on proving the cut elimination theorems only for proofs ending with the empty sequent. Before applying the reduction method on a given proof in order to reduce the cuts in it, Gentzen preforms a preparatory step in which he eliminates all appearances of the induction rule from the end-piece of the proof[14] . The elimination of the induction rule is done by replacing a complete induction up to a specific natural number by a corresponding number of structural inference rules. In order to make this replacement he first replaces all free variables which are not used as eigenvariables in the end-piece of the proof by constants.

[14] The end-piece of a proof consists of all the sequents of the proof encountered if we ascend each path starting from the end-sequent and stopping when we arrive to an operational inference rule. Thus the lower sequent of this inference rule belongs to the end-piece, but its upper sequents do not.

The transformation is done in the following way. Assume that within an end-piece we have the following segment

$$\dfrac{\overset{\vdots\; P'}{\Gamma, \psi\left\{\frac{a}{x}\right\} \Rightarrow \Delta, \psi\left\{\frac{s(a)}{x}\right\}}}{\Gamma, \psi\left\{\frac{0}{x}\right\} \Rightarrow \Delta, \psi\left\{\frac{t}{x}\right\}}$$

where P' is the sub-proof ending with $\Gamma, \psi\left\{\frac{a}{x}\right\} \Rightarrow \Delta, \psi\left\{\frac{s(a)}{x}\right\}$. Since all free variables were eliminated, t is a closed term and hence there is a term $s(\ldots(s(0))$ such that $\Rightarrow s(\ldots(s(0)) = t$ is provable in PA without essential cuts or induction. Therefore, there is also a proof Q of $\psi(s(\ldots(s(0))) \Rightarrow \psi(t)$ without essential cuts or induction. Let $P'\,(b)$ be the proof which is obtained from P' by replacing a by b throughout the proof.

Then an occurrence of the induction rule is replaced by:

$$\dfrac{\dfrac{\overset{\vdots\; P'(0)}{\Gamma, \psi\left\{\frac{0}{x}\right\} \Rightarrow \Delta, \psi\left\{\frac{s(0)}{x}\right\}} \quad \overset{\vdots\; P'(s(0))}{\Gamma, \psi\left\{\frac{s(0)}{x}\right\} \Rightarrow \Delta, \psi\left\{\frac{s(s(0))}{x}\right\}}}{\Gamma, \psi\left\{\frac{0}{x}\right\} \Rightarrow \Delta, \psi\left\{\frac{s(s(0))}{x}\right\}} \qquad \overset{\vdots\; P'(s(s(0)))}{\Gamma, \psi\left\{\frac{s(s(0))}{x}\right\} \Rightarrow \Delta, \psi\left\{\frac{s(s(s(0)))}{x}\right\}}}{\Gamma, \psi\left\{\frac{0}{x}\right\} \Rightarrow \Delta, \psi\left\{\frac{s(s(s(0)))}{x}\right\}}$$

We continue applying these consecutive cuts up to the sequent $\Gamma, \psi\left\{\frac{0}{x}\right\} \Rightarrow \Delta, \psi\left\{\frac{s(\ldots(s(0))}{x}\right\}$. Then we use one more cut on the sequent $\psi(s(\ldots(s(0))) \Rightarrow \psi(t)$ to obtain a proof of $\Gamma, \psi\left\{\frac{0}{x}\right\} \Rightarrow \Delta, \psi\left\{\frac{t}{x}\right\}$.

So, why can't we use the same method for the TC-induction rule? The problem is that this transformation uses special features of the natural numbers which we generally do not have in TC_G. To see this, recall that the equivalent form of the PA_G induction rule in TC_G is:

$$\dfrac{\Gamma, \psi\left(x\right), s\left(x\right) = y \Rightarrow \Delta, \psi\left\{\frac{y}{x}\right\}}{\Gamma, \psi\left\{\frac{0}{x}\right\}, \left(TC_{x,y}\left(s\left(x\right) = y\right)\right)\left(0, t\right) \Rightarrow \Delta, \psi\left\{\frac{t}{x}\right\}}$$

This is a specific usage of the induction rule in TC_G where φ is taken to be $s\left(x\right) = y$. However, in the general case φ is an arbitrary formula. Thus given any two closed terms s and t, unlike in PA_G, we do not have a "built in" measure for the φ-distance between them. The path from s to t by φ-steps is unknown in advance, and moreover it does not have to be unique. We may have more than one path from s to t using φ-steps. For instance, if φ is $y > x$ then the path from '1' to '20' may be '1','3','17','20' or '1','20' and we have no way of knowing it in advance.

Unfortunately, it is presently unclear whether we have cut elimination in TC_G, or rather we are required do add some derivable rules to the system in order for it to admit cut elimination. It is also to be determined what kinds of useful fragments of TC_G do admit cut elimination in case TC_G does not. Thus, a possible way to overcome the last difficulty described above is to restrict the induction rule of TC_G by allowing only φ's of the form $y = t$ where $\{x\} = Fv\,[t]$. In this way we force a deterministic path of φ-steps between any two closed terms. Obviously, induction with this restriction still includes PA_G's induction rule. By allowing an application of the induction rule only if φ is of such form we may be able to mimic the procedure given by Gentzen and eliminate the occurrences of the induction rule from proofs of the empty sequent. Then, by following the same reduction steps is seems that we may be able to prove the consistency of this restricted TC_G. This will be left for further work.

We end by discussing one more proof-theoretic property of the system RTC_A - constructive consistency proof.

Definition 5.21. A system is said to be *consistent* if the empty sequent is not provable in it.

The system RTC_A is surly consistent, since the natural numbers is a model for it. However, there are good reasons to look for a syntactic consistency proof. Beside consistency, constructive consistency proofs of the type introduced by Gentzen[8], usually provide important additional information about a system and its complexity. In Gentzen's method, each system is assigned the least ordinal number needed for its constructive consistency proof. This provides a measure for a complexity of a system which is useful for comparing different proof systems.

The constructive consistency proof of PA_G entail that the ordinal number of PA_G is at most ε_0 (for background on ordinal numbers see [12]). Another theorem of Gentzen shows that it is exactly ε_0. Hence, from Theorem 5.20, we can conclude that the ordinal number of the system RTC_A is ε_0.

6 Further research

The next list is a collection of related issues which require further work:

- An observation given in [1] is that in the presence of the existential quantifier it will be possible to use an even simpler form of TC: $\left(\tilde{TC}_{x,y}\varphi\right)$, which has the same meaning as $\left(TC_{x,y}\varphi\right)(x,y)$. In this kind of system we might not be able to substitute free variables by terms.

- What is the expressive power of ancestral logic if we disallow nesting of the TC operator. Will such a system suffice for formalizing mathematics?

- Characterizing the class of all first-order formulas whose transitive closure is first-order definable.

- In chapter 5 it was shown that the system TC_G does not captures all the intuitive properties of the TC operator. A research task here is to find what other "simple" rule(s?) should be added to TC_G in order to make it somewhat complete.

- In chapter 5 the property of cut-elimination for TC_G was discussed. Further research is required in order to determine what kinds of useful fragments of TC_G do admit cut-elimination. One possible option (already mentioned) is to restrict the induction rule of TC_G by allowing only φ's of the form $y = t$ where $\{x\} = Fv\,[t]$. Another option is to find out what are the conditions on a formula φ and terms s, t so that there is a proof in TC_G for $\Rightarrow \left(TC_{x,y}\varphi\right)(s,t)$ without the induction rule, and then restrict the induction rule by those conditions .

References

[1] A. Avron, *Transitive closure and the Mechanization of Mathematics,* see [23], 149–171.

[2] P. Smith, *Ancestral Arithmetic and Isaacson's Thesis,* Analysis, Vol. 68, Issue 297 (2008),1–10.

[3] J. Myhill, *A derivation on number theory from ancestral theory*, Journal of Symbolic Logic, Vol. 17 (1952), 192–197.

[4] R.M. Martin, *A Homogeneous System for Formal Logic,* Journal of Symbolic Logic, Vol. 8, No. 1 (1943), 1-23.

[5] R.M. Martin, *A Note on Nominalism and Recursive Functions,* Journal of Symbolic Logic, Vol. 14, No. 1 (1949), 27-31.

[6] S. Shapiro, *Foundations without Foundationalism: A Case for Second-Order Logic*, Oxford University Press (1991), 173-262.

[7] R. Heck, *The logic of Frege's Theorem*, Festschrift for Crispin Wright. (2007).

[8] G. Gentzen, *Die Widerspruchsfreiheit der reinen Zahlentheorie*, Mathematische Annalen, vol. 112 (1936) ,81-102, 252-286.

[9] G. Takeuti, *Proof theory,* Amsterdam:North-Holland. (1987), 22-126.

[10] A. Troelstra and H. Schwichtenberg, *Basic Proof Theory*, Cambridge, 2nd ed. (2000), 92-146.

[11] D. Monk, *Mathematical logic*, Springer-Verlag, New York (1976), 488-504.

[12] W. Sierpinski, *Cardinal and ordinal numbers*,2nd ed.(1965), 269-365.

[13] W. Pohlers, *Proof theory,* Springer-Verlag, Berlin, Heidelberg (1989) .

[14] Cowles, J. (1979), *The relative expressive power of some logics extending first-order logic.* Journal of Symbolic Logic, (1944), 129-46.

[15] B. Dreben and W.Goldfarb, *The Decision Problem: Solvable Classes of Quantificational Formulas*, Addison-Wesley Publishing Company (1979), section 8.3.

[16] L. Henkin, *Some remarks on infinitely long formulas*, Infinitistic methods, Pergamon Press, New York, and Polish Scientific Publishers, Warsaw, (1961), 167-183.

[17] D. Isaacson, *Arithmetical truth and hidden higher-order concepts*, In Logic Colloquium (1985), ed. The Paris Logic Group, Amsterdam: North-Holland, (1987), 147–69.

[18] W. Quine, *Mathematical Logic,* (rev. ed.), Cambridge (1951), 253-262.

[19] A. Blass and Y. Gurevich, *Henkin Quantifiers and complete problems*, Annals of pure and applied logic 32 (1986), 1-16.

[20] N. Immerman, *Languages That Capture Complexity Classes*, 15th symposium on theory of computing, association for computing machinery (1983), 347-354.

[21] J.R. Buchi, *On a decision method in restricted second order arithmetic,* Logic methodology and philosophy of science, Proceeding of the 1960 congress, Stanford university press (1962), 1-11.

[22] D. Gabbay, *What is a logical system?*, Oxford science publications, Clarendon press, Oxford (1994), 297-352.

[23] F. Kamareddine (ed.), *Thirty Five Years of Automating Mathematics*, Kluwer Academic Publishers (2003).