ON THE LOGICAL COMPLEXITY OF CYCLIC ARITHMETIC

ANUPAM DAS

Abstract. We study the logical complexity of proofs in cyclic arithmetic (CA), as introduced by Simpson in [Sim17], in terms of quantifier alternations of formulae occurring. Writing $\mathrm{C}\Sigma_n$ for (the logical consequences of) cyclic proofs containing only Σ_n formulae, our main result is that $\mathrm{I}\Sigma_{n+1}$ and $\mathrm{C}\Sigma_n$ prove the same Π_{n+1} theorems, for $n \geq 0$. Furthermore, due to the 'uniformity' of our method, we also show that CA and Peano Arithmetic (PA) proofs of the same theorem differ only elementarily in size.

The inclusion $\mathrm{I}\Sigma_{n+1} \subseteq \mathrm{C}\Sigma_n$ is obtained by proof theoretic techniques, relying on normal forms and structural manipulations of PA proofs. It improves upon the natural result that $\mathrm{I}\Sigma_n \subseteq \mathrm{C}\Sigma_n$.

The converse inclusion, $\mathrm{C}\Sigma_n \subseteq \mathrm{I}\Sigma_{n+1}$, is obtained by calibrating the approach of [Sim17] with recent results on the reverse mathematics of Büchi's theorem [KMPM16], and specialising to the case of cyclic proofs.

These results improve upon the bounds on proof complexity and logical complexity implicit in [Sim17] and [BT17b].

1. Introduction

Cyclic and *non-wellfounded* proofs have been studied by a number of authors as an alternative to proofs by induction. This includes cyclic systems for fragments of the modal μ-calculus, e.g. [NW96, SD03, DHL06, DBHS16, Dou17, AL17], structural proof theory for logics with fixed-points, e.g. [San02, FS13, For14, BDS16], (automated) proofs of program termination in separation logic, e.g. [BBC08, BDP11, RB17] and, in particular, cyclic systems for first-order logic with inductive definitions, e.g. [Bro05, Bro06, BS07, BS11]. Due to the somewhat implicit nature of invariants they define, cyclic proof systems can be advantageous for metalogical analysis, for instance offering better algorithms for proof search, e.g. [BGP12, DP17].

Cyclic proofs may be seen as more intuitively analogous to proofs by 'infinite descent' rather than proofs by induction (see, e.g., [Sim17]); this subtle difference is enough to make inductive invariants rather hard to generate from cyclic proofs. Indeed it was recently shown that simulating cyclic proofs using induction is not possible for some sub-arithmetic languages [BT17a], although becomes possible once arithmetic reasoning is available [Sim17, BT17b].

Cyclic arithmetic was proposed as a general subject of study by Simpson in [Sim17]. Working in the language of arithmetic, it replaces induction by non-wellfounded proofs with a certain 'fairness' condition on the infinite branches. The advantage of this approach to infinite proof theory as opposed to, say, infinite well-founded proofs via an ω-rule (see, e.g., [Sch77]), is that it admits a notion of *finite proofs*: those that have only finitely many distinct subproofs, and so may be represented by a finite (possibly cyclic) graph.

Date: February 19, 2018.

Cyclic arithmetic itself is to cyclic proofs what Peano arithmetic is to traditional proofs. It provides a general framework in which many arguments can be interpreted and/or proved in a uniform manner, and this is one reason why it is arguably a worthy subject of study. This is already clear from, say, the results of [BT17b], where the study of cyclic proofs for pure first-order logic with inductive definitions relied on an underlying arithmetic framework. We make some more comments on this in Sect. 8.

Contribution. In [Sim17], Simpson shows that Peano Arithmetic (PA) (i.e. with induction) is able to simulate cyclic reasoning by proving the soundness of the former in the latter. (The converse result is obtained much more easily.) Nonetheless, several open questions remain from [Sim17], concerning constructivity, normalisation, logical complexity and proof complexity for cyclic and non-wellfounded proofs.

In this work we address the *logical complexity* and *proof complexity* of proofs in Cyclic Arithmetic (CA), as compared to PA. Namely, we study how quantifier alternation of proofs in one system compares to that in the other, and furthermore how the size of proofs compare. Writing $C\Sigma_n$ for (the logical consequences of) cyclic proofs containing only Σ_n formulae, we show, for $n \geq 0$:

(1) $I\Sigma_{n+1} \subseteq C\Sigma_n$ over Π_{n+1} theorems (Sect. 4, Thm. 13).
(2) CA and PA proofs of the same theorem differ only elementarily in size (Sect. 6, Thm. 26).
(3) $C\Sigma_n \subseteq I\Sigma_{n+1}$ over all theorems (Sect. 7, Thm. 28).

(1) is obtained by proof theoretic techniques, relying on normal forms and structural manipulations of Peano Arithmetic proofs. It improves upon the natural result that $I\Sigma_n \subseteq C\Sigma_n$, although induces a non-elementary blowup in the size of proofs. (3) is obtained via a certain 'uniformisation' of the approach of [Sim17]. In particular, by specialising the key intermediate results to the case of cyclic proofs, we are able to extract small PA proofs of some required properties of infinite word automata from analogous ones in 'second-order' (SO) arithmetic. Finally, (3) is obtained by calibrating the argument of (2) with recent results on the reverse mathematics of Büchi's theorem [KMPM16], allowing us to bound the logical complexity of proofs in the simulation. Together, these results characterise the logical and proof complexity theoretic strength of cyclic proofs in arithmetic, resolving the questions (ii) and (iii), Sect. 7 of [Sim17].

This paper is structured as follows. In Sects. 2 and 3 we introduce some preliminaries on Peano Arithmetic, proof theory, cyclic proofs and automaton theory. In Sect. 4 we present (1). We introduce some SO theories of arithmetic in Sect. 5 that are conservative over the fragments of PA we need in order to conduct some of the intermediate arguments on infinite word automata. In Sect. 6 we present (2), and in Sect. 7 we adapt the argument to obtain (3). We conclude with some further remarks and perspectives in Sect. 8, including a comparison with the results of [Sim17] and [BT17b].

2. Preliminaries on first-order arithmetic proof theory

We present only brief preliminaries, but the reader is encouraged to consult, e.g., [Bus98] for a more thorough introduction to first-order arithmetic. We work in first-order (FO) logic with equality, with variables written x, y, z etc., terms written s, t, u etc., and formulae written φ, ψ etc., construed over $\{\neg, \vee, \wedge, \exists, \forall\}$.

We will usually assume formulae are in **De Morgan normal form**, with negation restricted to atomic formulae. Nonetheless, we may write $\neg\varphi$ for the De Morgan 'dual' of φ, defined as follows:

$$\neg\neg\varphi := \varphi \qquad \begin{array}{rcl} \neg(\varphi \wedge \psi) & := & \neg\varphi \vee \neg\psi \\ \neg(\varphi \vee \psi) & := & \neg\varphi \wedge \neg\psi \end{array} \qquad \begin{array}{rcl} \neg\forall x.\varphi & := & \exists x.\neg\varphi \\ \neg\exists x.\varphi & := & \forall x.\neg\varphi \end{array}$$

We also write $\varphi \supset \psi$ for $\neg\varphi \vee \psi$ and $\varphi \equiv \psi$ for $(\varphi \supset \psi) \wedge (\psi \supset \varphi)$.

Following [Sim17], the language of **arithmetic** is formulated as $\{0, \mathsf{s}, +, \times, <\}$, and a **theory** is a set T of closed formulae over this language. We write $T \vdash \varphi$ if φ is a logical consequence of T. We write $T_1 \subseteq T_2$ if $T_1 \vdash \varphi$ implies $T_2 \vdash \varphi$, and $T_1 = T_2$ if $T_1 \subseteq T_2$ and $T_2 \subseteq T_1$.

The theory of **Robinson arithmetic**, written Q, is axiomatised as follows:

i) $\forall x, y, z.((x < y \wedge y < z) \supset x < z)$.
ii) $\forall x, y.(\neg\, x < y \ \vee \neg\, y < x)$.
iii) $\forall x, y.(x < y \vee x = y \vee y < x)$.
iv) $\forall x.\ \neg\, x < 0$.
v) $\forall x, y.(x < y \supset \mathsf{s}x < \mathsf{s}y)$.
vi) $\forall x.\ x < \mathsf{s}x$.
vii) $\forall x, y.(\neg x < y \vee \neg y < \mathsf{s}x)$.
viii) $\forall x.\ x + 0 = x$.
ix) $\forall x, y.\ x + \mathsf{s}y = \mathsf{s}(x + y)$.
x) $\forall x.\ x \cdot 0 = 0$.
xi) $\forall x.\ x \cdot \mathsf{s}y = xy + x$.
xii) $\forall x.(0 < x \supset \exists y < x.\ x = \mathsf{s}y)$.

Notice that, above and elsewhere, we may write $\cdot$ instead of $\times$ in terms, or even omit the symbol altogether, and we assume it binds more strongly than $+$. We also write $\forall x \leq t.\varphi$ and $\exists x \leq t.\varphi$ as abbreviations for $\forall x.(x \leq t \supset \varphi)$ and $\exists x.(x \leq t \wedge \varphi)$ resp. Formulae with only such quantifiers are called **bounded**.

Definition 1 (Arithmetical hierarchy)**.** For $n \geq 0$, we define:

- $\Delta_0 = \Pi_0 = \Sigma_0$ is the class of bounded formulae.
- Σ_{n+1} is the class of formulae of the form $\exists \vec{x}.\varphi$, where $\varphi \in \Pi_n$.
- Π_{n+1} is the class of formulae of the form $\forall \vec{x}.\varphi$, where $\varphi \in \Sigma_n$.

We say that a sequent is Σ_n (or Π_n) if it contains only Σ (respectively Π_n) formulae.

Notice in particular that, due to De Morgan normal form, if $\varphi \in \Sigma_n$ then $\neg\varphi \in \Pi_n$ and vice-versa. In practice we often consider these classes of formulae up to logical equivalence. We say that a formula is in Δ_n (in a theory T) if it is equivalent to both a Σ_n and Π_n formula (provably in T).

Definition 2 (Arithmetic)**. Peano Arithmetic** (PA) is axiomatised by Q and the axiom schema of **induction**:

$$(\varphi(0) \wedge \forall x.(\varphi(x) \supset \varphi(\mathsf{s}x))) \supset \forall x.\varphi(x) \tag{1}$$

For a class of formulae Φ, we write Φ-IND for the set of induction axiom instances when $\varphi \in \Phi$ in (1). Furthermore, we write $\mathrm{I}\Phi$ for the theory $\mathsf{Q} + \Phi$-IND.

The following is a classical result:

Proposition 3 (E.g. [Bus98])**.** $\mathrm{I}\Sigma_n = \mathrm{I}\Pi_n$*, for* $n \geq 0$*.*

$$\frac{}{\Gamma \Rightarrow \Delta, t = t}\, {\scriptstyle t=t} \qquad \frac{\Gamma(s,t) \Rightarrow \Delta(s,t)}{s = t, \Gamma(t,s) \Rightarrow \Delta(t,s)}\, {\scriptstyle s=t} \qquad \frac{\Gamma \Rightarrow \Delta}{\theta(\Gamma) \Rightarrow \theta(\Delta)}\, {\scriptstyle \theta\text{-}sub}$$

$$\frac{}{\Gamma, \varphi \Rightarrow \Delta, \varphi}\, {\scriptstyle id} \qquad \frac{\Gamma \Rightarrow \Delta, \varphi \quad \Gamma, \varphi \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\, {\scriptstyle cut} \qquad \frac{\Gamma \Rightarrow \Delta}{\Gamma, \Gamma' \Rightarrow \Delta, \Delta'}\, {\scriptstyle wk}$$

$$\frac{\Gamma, \varphi \Rightarrow \Delta \quad \Gamma, \psi \Rightarrow \Delta}{\Gamma, \varphi \vee \psi \Rightarrow \Delta} \quad \frac{\Gamma, \varphi_i \Rightarrow \Delta}{\Gamma, \varphi_0 \wedge \varphi_1 \Rightarrow \Delta} \quad \frac{\Gamma, \varphi[a/x] \Rightarrow \Delta}{\Gamma, \exists x.\varphi \Rightarrow \Delta} \quad \frac{\Gamma, \varphi[t/x] \Rightarrow \Delta}{\Gamma, \forall x.\varphi \Rightarrow \Delta}$$

$$\frac{\Gamma \Rightarrow \Delta, \varphi \quad \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi} \quad \frac{\Gamma \Rightarrow \Delta, \varphi_i}{\Gamma \Rightarrow \Delta, \varphi_0 \vee \varphi_1} \quad \frac{\Gamma \Rightarrow \Delta, \varphi[a/x]}{\Gamma \Rightarrow \Delta, \forall x.\varphi} \quad \frac{\Gamma \Rightarrow \Delta, \varphi[t/x]}{\Gamma \Rightarrow \Delta, \exists x.\varphi}$$

FIGURE 1. Sequent calculus for FO logic with equality.

2.1. **A sequent calculus presentation of PA.** We will work with a standard sequent calculus presentation of FO logic, as in [Sim17], given in Fig. 1, where $i \in \{0, 1\}$ and a is fresh, i.e. does not occur free in the lower sequent. Two important considerations are that we work with cedents as *sets*, i.e. there is no explicit need for contraction rules, and that we have explicit **substitution** and **equality** rules. In the θ-*sub* rule the 'substitution' θ is a mapping from variables to terms, which is extended in the natural way to cedents. Substitution is important for the definition of a cyclic arithmetic proof in the next section, but does not change provability in usual proofs. While it can have an effect on proof complexity, this is only up to a polynomial factor once quantifiers are present.

The sequent calculus for Q is obtained from the FO calculus in the language of arithmetic by adding Δ_0 initial sequents for each instantiation of an axiom of Q by terms. For theories extending Q by induction axioms, we add the induction rule:

$$\frac{\Gamma \Rightarrow \varphi(0), \Delta \quad \Gamma, \varphi(a) \Rightarrow \varphi(\mathsf{s}a), \Delta}{\Gamma \Rightarrow \varphi(t), \Delta}\, {\scriptstyle ind}$$

Here we require a to not occur free in the lower sequent. Notice that this satisfies the subformula property, in the 'wide' sense of FO logic, i.e. up to substitution of variables by terms. For fragments of PA with induction axioms of bounded logical complexity, we also use the *bounded quantifier rules*:

$$\frac{\Gamma, a < s, \varphi(a) \Rightarrow \Delta}{\Gamma, \exists x < s.\varphi(x) \Rightarrow \Delta} \qquad \frac{\Gamma \Rightarrow \Delta, \varphi(t)}{\Gamma, t < s \Rightarrow \Delta, \exists x < s.\varphi(x)}$$

$$\frac{\Gamma, a < s \Rightarrow \Delta, \varphi(a)}{\Gamma \Rightarrow \Delta, \forall x < s.\varphi(x)} \qquad \frac{\Gamma, \varphi(t) \Rightarrow \Delta}{\Gamma, t < s, \forall x < s.\varphi(x) \Rightarrow \Delta}$$

In all cases the eigenvariable a occurs only as indicated,

The following normalisation result is well-known in the proof theory of arithmetic, and will be one of our main proof-theoretic tools throughout this work:

Theorem 4 (Free-cut elimination, e.g. [Bus98])**.** *Let $\mathcal{S}$ be a sequent system extending FO by the induction rule and some initial sequents closed under substitution. Then any $\mathcal{S}$-proof can be effectively transformed into one of the same conclusion containing only (substitution instances of) subformulae of the conclusion, an induction formula or a formula in an intial sequent.*

Naturally, this applies to the various fragments of PA that we consider. In particular, notice that a proof in $\mathrm{I}\Sigma_n$ or $\mathrm{I}\Pi_n$ of Σ_n or Π_n sequents, resp., contains just Σ_n or Π_n formulae, resp.

A slight issue that will be relevant later in Sect. 4 is that we have not defined Σ_n and Π_n as being syntactically closed under positive Boolean combinations, even if semantically we know that they are. In fact, this does not cause a problem for the result above, since we can always prenex 'on the fly' in a proof by cutting against appropriate derivations. For instance, in a proof, a step of the form,

$$\wedge\,\frac{\Gamma \Rightarrow \Delta, \forall x.\varphi \quad \Gamma \Rightarrow \Delta, \forall y.\psi}{\Gamma \Rightarrow \Delta, \forall x.\varphi \wedge \forall y.\psi}$$

may be locally replaced by a derivation of the form:

$$cut\,\frac{\Gamma \Rightarrow \Delta, \forall x.\varphi \qquad cut\,\dfrac{\Gamma \Rightarrow \Delta, \forall y.\psi \quad \overline{\overline{\forall x.\varphi, \forall y.\psi \Rightarrow \forall x, y.(\varphi \wedge \psi)}}}{\Gamma, \forall y.\psi \Rightarrow \Delta, \forall x, y.(\varphi \wedge \psi)}}{\Gamma \Rightarrow \Delta, \forall x, y.(\varphi \wedge \psi)}$$

In a similar way we will often assume that a 'block' of existential or universal quantifiers is coded by a single quantifier, using pairings and Gödel β functions, whose basic properties are all formalisable already in $\mathrm{I}\Delta_0$ (see, e.g., [Bus98]).

3. Preliminaries on cyclic arithmetic and automata

Before presenting 'cyclic arithmetic', we will present the general notion of non-wellfounded proofs in arithmetic, from [Sim17].

Definition 5. A **preproof** is a possibly infinite binary tree labeled by sequents in a locally correct manner in the calculus for Q. As in [Sim17], we treat inference steps as nodes of the tree and sequents as edges. A preproof is **regular** if it has only finitely many distinct subtrees or, equivalently, if it is the unfolding of a finite directed graph, possibly with cycles.

The following notions are directly from Dfns. 1 and 2 in [Sim17]:

Definition 6 (Precursors, traces, ∞-proofs)**.** Let $(\Gamma_i \Rightarrow \Delta_i)_{i\geq 0}$ be an infinite branch through a preproof. For terms t, t' we say that t' is a **precursor** of t at i if one of the following holds:

- $\Gamma_i \Rightarrow \Delta_i$ is the conclusion of a θ-*sub*-step and $t = \theta(t')$.
- $\Gamma_i \Rightarrow \Delta_i$ is the conclusion of an $t_1 = t_2$-step, and $t' = t[t_1/t_2, t_2/t_1]$.
- $\Gamma_i \Rightarrow \Delta_i$ is the conclusion of any other rule and $t = t'$.

A **trace** along $(\Gamma_i \Rightarrow \Delta_i)_{i\geq 0}$ is a sequence $(t_i)_{i\geq n}$, for some $n \geq 0$, such that whenever $i \geq n$ the term t_i occurs in $\Gamma_i \Rightarrow \Delta_i$ and,

(a) t_{i+1} is a precursor of t_i at i; or
(b) the atomic formula $t_{i+1} < t$ occurs in Γ_{i+1}, where t is a precursor of t_i at i.

When (b) holds, we say that the trace **progresses** at $i + 1$.

An **∞-proof** is a preproof such that any infinite branch has a trace that progresses infinitely often. If it is regular then we simply call it a **cyclic proof**. CA is the class of cyclic proofs built over initial sequents corresponding to Q.

The reader may consult [Sim17] for several examples of ∞-proofs. Notably, ∞-proofs are *complete* for the standard model $\mathbb{N}$ (Thm. 4, [Sim17]). On the other hand, soundness of ∞-proofs for various logics with respect to a standard model is now a well-known result [Bro06, BS11], but for the current setting it was stated and proved in [Sim17]. We recall the proof since we will have to formalise a variant of it in Sect. 6, and also since the quantifier case in the argument of [Sim17] is omitted, whereas this subtlety will need some consideration when it is formalised.

Proposition 7 (Soundness)**.** *If π is an ∞-proof of φ, then $\mathbb{N} \vDash \varphi$.*

Proof sketch. Suppose otherwise, i.e. $\mathbb{N} \vDash \neg\varphi$. We will inductively construct an infinite branch $(\Gamma_i \Rightarrow \Delta_i)_{i \geq 0}$ of π and associated assignments ρ_i of natural numbers to each sequent's free variables, such that $\mathbb{N}, \rho_i \nvDash \Gamma_i \Rightarrow \Delta_i$. Assuming φ is closed (by taking its universal closure), we set $\Gamma_0 \Rightarrow \Delta_0$ to be $\Rightarrow \varphi$ and $\rho_0 = \varnothing$.

Each step except for substitution, $\forall$-right and $\exists$-left constitutes a true implication, so if $\mathbb{N}, \rho_i \nvDash \Gamma_i \Rightarrow \Delta_i$ then ρ_i also must not satisfy one of its premisses. We may thus choose one such premiss as $\Gamma_{i+1} \Rightarrow \Delta_{i+1}$ and set $\rho_{i+1} = \rho_i$.

If $\Gamma_i \Rightarrow \Delta_i$ concludes a θ-*sub*-step, we may set $\rho_{i+1} = \theta(\rho)$. If $\Gamma_i \Rightarrow \Delta_i$ concludes a $\forall$-right step, let $\forall x.\varphi$ be the principal formula and assume x does not occur free in the conclusion. Since $\mathbb{N}, \rho_i \nvDash \Gamma_i \Rightarrow \Delta_i$, we must have that $\mathbb{N}, \rho_i \vDash \exists x.\neg\varphi$. We choose a value $k \in \mathbb{N}$ witnessing this existential and set $\rho_{i+1} = \rho_i \cup \{x \mapsto k\}$. The $\exists$-left case is dealt with similarly.

This infinite branch must have an infinitely progressing trace, say $(t_i)_{i \geq n}$, by the definition of ∞-proof. However notice that, for $i \geq n$, $\rho_i(t_i) \geq \rho_{i+1}(t_{i+1})$ and, furthermore, at a progress point along the trace, $\rho_i(t_i) > \rho_{i+1}(t_{i+1})$. Thus, $(\rho_i(t_i))_{i \geq n}$ is a monotone decreasing sequence of natural numbers that does not converge, contradicting the fact that $\mathbb{N}$ is well-ordered. $\square$

Later, in Sect. 6, we will use the fact that the choices for generating an invalid branch in the proof above can be made *uniformly* in an arithmetic setting.

3.1. **Defining $\mathrm{C}\Sigma_n$.** Simpson proposes in [Sim17] to study systems of cyclic proofs containing only Σ_n formulae, and to compare such systems to $\mathrm{I}\Sigma_n$. This is rather pertinent in light of the free-cut elimination result we stated, Thm. 4: any $\mathrm{I}\Sigma_n$-proof of a Σ_n-sequent can be assumed to contain just Σ_n formulae possibly at a non-elementary cost in proof size, whence the comparison. However, in order to be able to admit routine derivations of more complex formulae, e.g. the Σ_{n+1} law of excluded middle or the universal closure of a Σ_n sequent, we will close this notion under *logical consequence*.

Definition 8. Let Φ be closed under subformulae and substitution. $\mathrm{C}\Phi$ is the first-order theory axiomatised by the universal closures of conclusions of cyclic proofs containing only Φ-formulae.

Notice that, by the free-cut elimination result, Thm. 4, and the subformula property, any $\mathrm{C}\Sigma_n$ proofs of Σ_n-sequents contain only Σ_n-sequents anyway, without loss of generality. This more 'robust' definition allows us to easily compare fragments of cyclic arithmetic. For instance, we have the following:

Proposition 9. $\mathrm{C}\Sigma_n = \mathrm{C}\Pi_n$, *for $n \geq 0$.*

Proof sketch. For the left-right inclusion, replace each Σ_n sequent $\vec{p}, \Gamma \Rightarrow \Delta$ with the sequent $\vec{p}, \overline{\Delta} \Rightarrow \overline{\Gamma}$, where $\overline{\Gamma}$ and $\overline{\Delta}$ contain the De Morgan dual formulae of Γ

and Δ resp. and any atomic formulae are amongst $\vec{p}$. Any traces will be preserved, and the proof can be made correct by adding some extra logical steps. The converse implication is proved in the same way. $\square$

Using a standard technique, e.g. from [Bro06], we also can rather simply show the following result:

Proposition 10. $\mathrm{I}\Sigma_n \subseteq \mathrm{C}\Sigma_n$, *for* $n \geq 0$.

Note that, while a similar result was given in [Sim17], that argument rather shows that $\mathrm{I}\Sigma_n \subseteq \mathrm{C}\Sigma_{n+1}$.

Proof sketch. Given an $\mathrm{I}\Sigma_n$ proof, we simply simulate every local inference step, the only nontrivial case being an induction step:

$$
{\scriptstyle ind}\ \dfrac{\Gamma \Rightarrow \varphi(0), \Delta \quad \Gamma, \varphi(a) \Rightarrow \varphi(\mathsf{s}a), \Delta}{\Gamma \Rightarrow \varphi(t), \Delta}
$$

This is simulated by the following cyclic derivation,

$$
{\scriptstyle sub}\ \dfrac{{\scriptstyle cut}\ \dfrac{{\scriptstyle =}\ \dfrac{\Gamma \Rightarrow \varphi(0), \Delta}{0 = b, \Gamma \Rightarrow \varphi(b), \Delta} \qquad \dfrac{\dfrac{\dfrac{{\scriptstyle cut}\ \dfrac{{\scriptstyle sub}\ \dfrac{{\scriptstyle cut}\ \dfrac{\vdots}{\Gamma \Rightarrow \varphi(b), \Delta}\ \bullet}{\Gamma \Rightarrow \varphi(a), \Delta} \qquad \Gamma, \varphi(a) \Rightarrow \varphi(\mathsf{s}a), \Delta}{\underline{a < b}, \Gamma \Rightarrow \varphi(\mathsf{s}a), \Delta}}{b = \mathsf{s}a, \Gamma \Rightarrow \varphi(b), \Delta}}{0 < b, \Gamma \Rightarrow \varphi(b), \Delta}}{\Gamma \Rightarrow \varphi(b), \Delta}\ \bullet}{\Gamma \Rightarrow \varphi(t), \Delta}
$$

where we have written $\bullet$ to mark roots of identical subtrees. Any infinite branch that does not have a tail in the proofs of the two premisses of *ind* must hit $\bullet$ infinitely often. Therefore they admit an infinitely progressing trace alternating between a and b, with the progress point underlined above. $\square$

Notice that, e.g. in the simulation of induction above, traces need not be connected in the graph of 'ancestry' of a proof, cf. [Bus98]. This deviates from other settings where it is *occurrences* that are tracked, rather than terms [DBHS16, BDS16, Dou17].

3.2. **Büchi automata: checking correctness of cyclic proofs.** A cyclic preproof can be effectively checked for correctness by reduction to the inclusion of 'Büchi automata', yielding a **PSPACE** bound. As far as the author is aware, this is the best known upper bound, although no corresponding lower bound is known. As we will see later in Sect. 6, this is one of the reasons why we cannot hope for a 'polynomial simulation' of cyclic proofs in a normal proof system, and so why elementary simulations are more pertinent.

Definition 11. A **nondeterministic Büchi automaton** (NBA) $\mathcal{A}$ is a tuple (A, Q, δ, q_0, F) where: A is a finite set, called the **alphabet**, Q is a finite set of **states**, $\delta \subseteq (Q \times A) \times Q$ is the **transition relation**, $q_0 \in Q$ is the **initial** state, and $F \subseteq Q$ is the set of **final** or **accepting** states. We say that $\mathcal{A}$ is **deterministic** (a DBA) if δ is a function $Q \times A \to Q$. A 'word' $(a_i)_{i \geq 0} \in A^\omega$ is **accepted** or

recognised by $\mathcal{A}$ if there is a sequence $(q_i)_{i\geq 0} \in Q^\omega$ such that: for each $i \geq 0$, $(q_i, a_i, q_{i+1}) \in \delta$, and for infinitely many i we have $q_i \in F$.

From a cyclic preproof π we can easily define two automata, say $\mathcal{A}_b^\pi$ and $\mathcal{A}_t^\pi$,[1] respectively accepting just the branches and just the branches with infinitely progressing traces. See [Sim17] for a construction of $\mathcal{A}_t^\pi$. We point out that $\mathcal{A}_b^\pi$ is essentially just the dependency graph of π with all states final, and so is in fact deterministic. Technically the transition relation here is not total, but this can be 'completed' in the usual way by adding a non-final 'sink' state for any outstanding transitions. We thus state the now well-known 'correctness condition' for cyclic proofs:

Proposition 12 ([Sim17])**.** *A cyclic preproof π is a ∞-proof if and only if $\mathcal{L}(\mathcal{A}_b^\pi) \subseteq \mathcal{L}(\mathcal{A}_t^\pi)$.*

4. A translation from $\mathrm{I}\Sigma_{n+1}$ to $\mathrm{C}\Sigma_n$, over Π_{n+1}-theorems

We show in this section our first result, that cyclic proofs containing only Σ_n-formulae are enough to simulate $\mathrm{I}\Sigma_{n+1}$:

Theorem 13. $\mathrm{I}\Sigma_{n+1} \subseteq \mathrm{C}\Sigma_n$*, over* Π_{n+1} *theorems, for* $n \geq 0$.

One example of such logical power in cyclic proofs was given in [Sim17], in the form of $\mathrm{C}\Sigma_1$-proofs of the totality of the Ackermann-Péter function: $\mathrm{I}\Sigma_1$ only proves the totality of the primitive recursive functions [Par72].

To prove the theorem above, we will rather work in $\mathrm{I}\Pi_{n+1}$, cf. Prop. 3, since the exposition is more intuitive. We first prove the following intermediate lemma.

Lemma 14. *Let π be a $\mathrm{I}\Pi_{n+1}$ proof, containing only Π_{n+1} formulae, with conclusion,*

$$\Gamma, \forall x_1.\varphi_1, \dots, \forall x_l.\varphi_l \Rightarrow \Delta, \forall y_1.\psi_1, \dots, \forall y_m.\psi_m \tag{2}$$

where $\Gamma, \Delta, \varphi_i, \psi_j$ are Σ_n and $\vec{x}, \vec{y}$ occur only in $\vec{\varphi}, \vec{\psi}$ respectively. Then there is a $\mathrm{C}\Sigma_n$ derivation $\lceil \pi \rceil$ of the form:

$$\begin{array}{c} \{\Gamma \Rightarrow \Delta, \varphi_i\}_{i \leq l} \\ \lceil \pi \rceil \\ \Gamma \Rightarrow \Delta, \psi_1, \dots, \psi_m \end{array}$$

Proof sketch. We proceed by induction on the structure of π. Notice that we may assume that any Π_{n+1} formulae occurring have just a single outermost $\forall$ quantifier, by interpreting arguments as pairs and using Gödel's β functions. This introduces only cuts on formulae of the same form.

We henceforth write $\vec{\psi}$ for $\psi_1, \dots, \psi_m$ and, as an abuse of notation, $\forall \vec{x}.\vec{\varphi}$ and $\forall \vec{y}.\vec{\psi}$ for $\forall x_1.\varphi_1, \dots, \forall x_l.\varphi_l$ and $\forall y_1.\psi_1, \dots, \forall y_m.\psi_m$ respectively. (Notice that this is a reasonable abuse of notation, since the $\forall$s can be prenexed outside a conjunction or disjunction already in pure FO logic.) We sketch only the interesting inductive cases here.

Propositional logical steps are easily dealt with, relying on invertibility and cuts, with possible structural steps. Importantly, due to the statement of the lemma,

[1]These are rather called B_p and B_t respectively in [Sim17].

such steps apply to only Σ_n formulae. For instance, if π extends a proof π' by a $\wedge$-left step,

$$\frac{\Gamma, \chi_0, \chi_1, \forall\vec{x}.\vec{\varphi} \Rightarrow \Delta, \forall\vec{y}.\vec{\psi}}{\Gamma, \chi_0 \wedge \chi_1, \forall\vec{x}.\vec{\varphi} \Rightarrow \Delta, \forall\vec{y}.\vec{\psi}}$$

then we define $\lceil\pi\rceil$ as,

$$\left\{ cut\,\frac{\overline{\overline{\chi_0, \chi_1 \Rightarrow \chi_0 \wedge \chi_1}} \quad \Gamma, \chi_0 \wedge \chi_1 \Rightarrow \Delta, \varphi_i}{\Gamma, \chi_0, \chi_1 \Rightarrow \Delta, \varphi_i} \right\}_{i \leq l}$$

$$\lceil\pi'\rceil$$

$$\frac{\Gamma, \chi_0, \chi_1 \Rightarrow \Delta, \vec{\psi}}{\Gamma, \chi_0 \wedge \chi_1 \Rightarrow \Delta, \vec{\psi}}$$

and if π extends proofs π_0, π_1 by a $\wedge$-right step,

$$\frac{\Gamma, \forall\vec{x}.\vec{\varphi} \Rightarrow \Delta, \chi_0, \forall\vec{y}.\vec{\psi} \quad \Gamma, \forall\vec{x}.\vec{\varphi} \Rightarrow \Delta, \chi_1, \forall\vec{y}.\vec{\psi}}{\Gamma, \forall\vec{x}.\vec{\varphi} \Rightarrow \Delta, \chi_0 \wedge \chi_1, \forall\vec{y}.\vec{\psi}}$$

then we define $\lceil\pi\rceil$ as:

$$\left\{ cut\,\frac{\Gamma \Rightarrow \Delta, \chi_0 \wedge \chi_1, \varphi_i \quad \overline{\overline{\chi_0 \wedge \chi_1 \Rightarrow \chi_j}}}{\Gamma \Rightarrow \Delta, \chi_j, \varphi_i} \right\}_{i \leq l}$$

$$\lceil\pi_j\rceil$$

$$\frac{\Gamma \Rightarrow \Delta, \chi_j, \vec{\psi}}{\Gamma \Rightarrow \Delta, \chi_0 \wedge \chi_1, \vec{\psi}}\,\forall j < 2$$

If π extends a proof π' by a thinning step,

$$wk\,\frac{\Gamma, \forall\vec{x}.\vec{\varphi} \Rightarrow \Delta, \forall\vec{y}.\vec{\psi}}{\Gamma', \Pi, \Gamma, \forall\vec{x}.\vec{\varphi} \Rightarrow \Delta, \forall\vec{y}.\vec{\psi}, \Delta', \forall\vec{z}.\vec{\chi}}$$

where $\Gamma', \Delta', \vec{\chi}$ are Σ_n and Π is Π_{n+1}, then we define $\lceil\pi\rceil$ as:

$$\{\Gamma', \Gamma \Rightarrow \Delta, \varphi_i, \Delta'\}_{i \leq l}$$

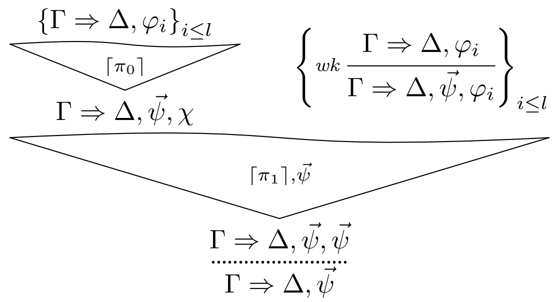
$$\Gamma', \lceil\pi'\rceil, \Delta'$$

$$wk\,\frac{\Gamma', \Gamma \Rightarrow \Delta, \vec{\psi}, \Delta'}{\Gamma', \Gamma \Rightarrow \Delta, \vec{\psi}, \Delta', \vec{\chi}}$$

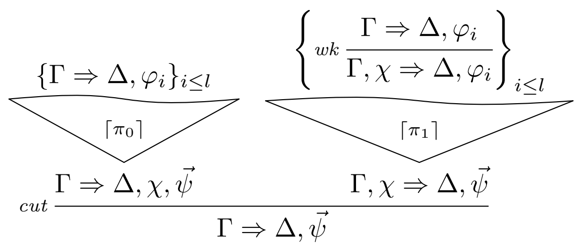
where $\Gamma', \lceil\pi'\rceil, \Delta'$ is obtained from $\lceil\pi'\rceil$ by prepending Γ' and appending Δ' to each sequent. For this we might need to rename some variables in π' so that eigenvariable conditions are preserved after the transformation. Notice that we are simply ignoring the extra premisses due to Π. If π extends proofs π_0, π_1 by a cut step on a Π_{n+1} formula,

$$\mathit{cut}\,\frac{\Gamma, \forall \vec{x}.\vec{\varphi} \Rightarrow \Delta, \forall \vec{y}.\vec{\psi}, \forall z.\chi(z) \quad \Gamma, \forall \vec{x}.\vec{\varphi}, \forall z.\chi(z) \Rightarrow \Delta, \forall \vec{y}.\vec{\psi}}{\Gamma, \forall \vec{x}.\vec{\varphi} \Rightarrow \Delta, \forall \vec{y}.\vec{\psi}}$$

then we define $\lceil \pi \rceil$ as:



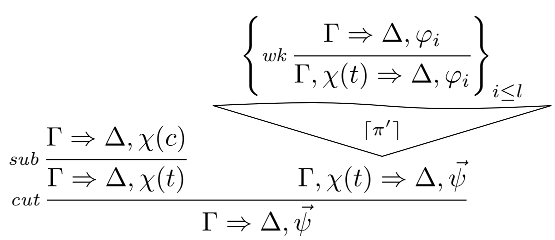
The final dotted 'contraction' step is implicit, since we treat cedents as sets. Again, we might need to rename some variables in π_1. If instead the cut formula were Σ_n, say χ, we would define $\lceil \pi \rceil$ as:



If π extends a proof π' by a $\forall$-left step,

$$\frac{\Gamma, \chi(t), \forall \vec{x}.\vec{\varphi} \Rightarrow \Delta, \forall \vec{y}.\vec{\psi}}{\Gamma, \forall z.\chi(z), \forall \vec{x}.\vec{\varphi} \Rightarrow \Delta, \forall \vec{y}.\vec{\psi}}$$

where $\forall z.\chi(z)$ is Π_{n+1}, we define $\lceil \pi \rceil$ as follows:



If π extends a proof π' by a $\forall$-right step,

$$\frac{\Gamma, \forall \vec{x}.\vec{\varphi} \Rightarrow \Delta, \forall \vec{y}.\vec{\psi}, \chi}{\Gamma, \forall \vec{x}.\vec{\varphi} \Rightarrow \Delta, \forall \vec{y}.\vec{\psi}, \forall z.\chi}$$

where $\forall z.\chi$ is Π_{n+1}, then we define $\lceil \pi \rceil$ as:

$$\begin{array}{c}
\left\{ {\scriptstyle wk}\,\dfrac{\Gamma \Rightarrow \Delta, \varphi_i}{\Gamma \Rightarrow \Delta, \varphi_i, \chi} \right\}_{i \leq l} \\
\lceil \pi' \rceil \\
\Gamma \Rightarrow \Delta, \vec{\psi}, \chi
\end{array}$$

Here we may need to again rename some variables in χ and apply substitutions at the end. If π extends a proof π' by a $\exists$-right step,

$$\dfrac{\Gamma, \forall \vec{x}.\vec{\varphi} \Rightarrow \Delta, \forall \vec{y}.\vec{\psi}, \chi(t)}{\Gamma, \forall \vec{x}.\vec{\varphi} \Rightarrow \Delta, \forall \vec{y}.\vec{\psi}, \exists z.\chi(z)}$$

where $\exists z.\chi(z)$ is Σ_n, then we define $\lceil \pi \rceil$ as:

$$\begin{array}{c}
\left\{ {\scriptstyle wk}\,\dfrac{\Gamma \Rightarrow \Delta, \varphi_i, \exists z.\chi(z)}{\Gamma \Rightarrow \Delta, \varphi_i, \chi(t), \exists z.\chi(z)} \right\}_{i \leq l} \\
\lceil \pi' \rceil, \exists z.\chi(z) \\
{\scriptstyle \exists}\,\dfrac{\Gamma \Rightarrow \Delta, \vec{\psi}, \chi(t), \exists z.\chi(z)}{\Gamma \Rightarrow \Delta, \vec{\psi}, \exists z.\chi(z)}
\end{array}$$

Again, some variables in π' might have to be renamed. Any other quantifier steps are dealt with routinely.

Finally, if π extends proofs π_0, π' by an induction step,

$${\scriptstyle ind}\,\dfrac{\Gamma, \forall \vec{x}.\vec{\varphi} \Rightarrow \Delta, \forall \vec{y}.\vec{\psi}, \forall z.\chi(0) \qquad \Gamma, \forall \vec{x}.\vec{\varphi}, \forall z.\chi(c) \Rightarrow \Delta, \forall \vec{y}.\vec{\psi}, \forall z.\chi(\mathsf{s}c)}{\Gamma, \forall \vec{x}.\vec{\varphi} \Rightarrow \Delta, \forall \vec{y}.\vec{\psi}, \forall x.\chi(t)}$$

we define $\lceil \pi \rceil$ to be the following cyclic proof,

$$\dfrac{
{\scriptstyle =}\,\dfrac{\begin{array}{c}\{\Gamma \Rightarrow \Delta, \varphi_i\}_{i \leq l} \\ \lceil \pi_0 \rceil \\ \Gamma \Rightarrow \Delta, \vec{\psi}, \varphi(0)\end{array}}{b = 0, \Gamma \Rightarrow \Delta, \vec{\psi}, \chi(d)}
\qquad
\dfrac{\begin{array}{c}{\scriptstyle sub}\,\dfrac{\dfrac{\vdots}{\Gamma \Rightarrow \Delta, \chi(d)}\,\bullet}{\Gamma \Rightarrow \Delta, \chi(c)} \quad \{\Gamma \Rightarrow \Delta, \varphi_i\}_{i \leq l} \\ \lceil \pi' \rceil \\ \underline{c < d}, \Gamma \Rightarrow \Delta, \vec{\psi}, \chi(\mathsf{s}c)\end{array}}{d = \mathsf{s}c, \Gamma \Rightarrow \Delta, \vec{\psi}, \chi(d)}
}{{\scriptstyle sub}\,\dfrac{\Gamma \Rightarrow \Delta, \vec{\psi}, \chi(d)}{\Gamma \Rightarrow \Delta, \vec{\psi}, \chi(t)}}\,\bullet$$

where we have written $\bullet$ to mark roots of identical subtrees. Notice that any branch hitting $\bullet$ infinitely often will have an infinitely progressing trace alternating between c and d, by the underlined progress point $c < d$: again, thanks to the eigenvariable conditions, we may rename variables in $\lceil \pi' \rceil$ so that no substitutions interfere with this trace. Otherwise an infinite branch has a tail that is already in $\lceil \pi' \rceil$ (or $\lceil \pi_0 \rceil$), and so has an inifinitely progressing trace by the inductive hypothesis. $\square$

The lemma above gives us a simple proof of the main result of this section:

Proof of Thm. 13. Suppose π is a $\mathrm{I}\Pi_{n+1}$ proof of a Σ_n sequent, under Prop. 3. By Thm. 4 we may assume that π contains only Π_{n+1} cuts, whence we may simply apply Lemma 14 to obtain the required result. Notice that, by 'de-generalising' the outer $\forall$ of a Π_{n+1} theorem, there are only Σ_n formulae in the conclusion of π, possibly with free variables, and so there are no assumption sequents after applying Lemma 14. $\square$

5. Some second-order theories and conservativity

Again, we give only brief preliminaries, but the reader is encouraged to consult the standard texts [Sim09, Hir14]. We now consider a two-sorted, or 'second-order' (SO), version of FO logic, with variables X, Y, Z, etc. ranging over sets of individuals, and new atomic formulae $t \in X$, sometimes written $X(t)$. We write Q_2 for an appropriate extension of Q by basic axioms governing sets (see, e.g., [Sim09] or [Hir14]), and write Σ^0_n, Π^0_n and Δ^0_n for the classes Σ_n, Π_n and Δ_n respectively, but now allowing free set variables to occur.

Definition 15. The **recursive comprehension** axiom schema is the following:

$$\Delta^0_1\text{-}\mathsf{CA} \ : \ \forall \vec{y}, \vec{Y}.(\forall x.(\varphi(x,\vec{y},\vec{Y}) \equiv \neg\psi(x,\vec{y},\vec{Y})) \supset \exists X.\forall x.(Xx \equiv \varphi(x)))$$

where φ, ψ are in Σ^0_1 and X does not occur free in φ or ψ. From here, the theory RCA_0 is defined as $\mathsf{Q}_2 + \Delta^0_1\text{-}\mathsf{CA} + \Sigma^0_1\text{-}\mathsf{IND}$.

Notice that there is an unfortunate coincidence of the notation CA for 'comprehension axiom' and 'cyclic arithmetic', but the context of use should always avoid any ambiguity.

Since we will always work in extensions of RCA_0, which prove the totality of primitive recursive functions, we will conservatively add function symbols for primitive recursive functions on individuals whenever we need them. We will also henceforth consider FO theories extended by 'oracles', i.e. uninterpreted set/predicate variables. For instance, we write $\mathrm{I}\Sigma_n(X)$ for the same class of proofs as $\mathrm{I}\Sigma_n$ but where the symbol X is allowed to occur as a predicate. The following result is an adaptation of well known conservativity results, e.g. as found in [Sim09, Hir14].

Proposition 16. $\mathsf{RCA}_0 + \Sigma^0_n\text{-}\mathsf{IND}$ *is conservative over* $\mathrm{I}\Sigma_n(X)$.

Proof sketch. First we introduce countably many fresh set symbols $X^{\vec{t},\vec{Y}}_{\varphi,\psi}$, indexed by Σ^0_1 formulae $\varphi(x,\vec{x},\vec{X}), \psi(x,\vec{x},\vec{X})$ with all free variables indicated, FO terms $\vec{t}$ with $|\vec{t}| = |\vec{x}|$ and SO variables $\vec{Y}$ with $|\vec{Y}| = |\vec{X}|$. These will serve as witnesses to the sets defined by comprehension. We replace the comprehension axioms by initial sequents of the form:

$$\frac{}{\forall x.(\varphi(x,\vec{t},\vec{Y}) \equiv \neg\psi(x,\vec{t},\vec{Y})), \varphi(t,\vec{t},\vec{Y}) \Rightarrow t \in X^{\vec{t},\vec{Y}}_{\varphi,\psi}}$$

$$\frac{}{\forall x.(\varphi(x,\vec{t},\vec{Y}) \equiv \neg\psi(x,\vec{t},\vec{Y})), t \in X^{\vec{t},\vec{Y}}_{\varphi,\psi} \Rightarrow \psi(t,\vec{t},\vec{Y})}$$

It is routine to show that these new initial sequents define a system equipotent to $\mathsf{RCA}_0 + \Sigma^0_n\text{-}\mathsf{IND}$.

Now we apply free-cut elimination, Thm. 4, to a proof in such a system and replace every positive occurrence of $t \in X^{\vec{t},\vec{Y}}_{\varphi,\psi}$ with $\varphi(t,\vec{t},\vec{Y})$, and every negative occurrence, i.e. an occurrence of $t \notin X^{\vec{t},\vec{Y}}_{\varphi,\psi}$ with $\psi(t,\vec{t},\vec{Y})$. (Recall here that we assume formulae are in De Morgan normal form.) The comprehension initial sequents are now just identities in the positive case, and purely logical theorems in the negative case. Any extraneous free set variables in induction steps (except X), e.g. Y, may be safely dealt with by replacing any atomic formula $s \in Y$ with $\top$. The resulting proof is in $\mathrm{I}\Sigma_n(X)$. □

It is worth pointing out that the transformation from a SO proof to a FO proof can yield a possibly non-elementary blowup in the size of proofs, due to, e.g., the application of (free-)cut elimination.

5.1. **Automaton theory for ω-languages in second-order arithmetic.** From now on we will be rather informal when talking about finite objects, e.g. automata, finite sequences, or even formulae. In particular, we may freely use such objects within object-level formulae, when in fact we are formally referring to their 'Gödel numbers'. Also, statements inside quotations, "-", will usually be (provably) recursive in any free variables occurring, i.e. Δ^0_1. This way quantifier complexity is safely measured by just the quantifiers outside quotations.

We may treat a set symbol X as a binary predicate by interpreting its argument as a pair and using Gödel's 'β functions' to primitively recursively extract its components. We use such predicates to encode sequences by interpreting $X(x,y)$ as "the x^{th} symbol of X is y". Axiomatically, this means we presume we have already the totality and determinism of X as a binary relation. Formally, for a set S and a set symbol X treated as a binary predicate, we will write $X \in S^\omega$ for the conjunction of the following two formulae,

$$\forall x. \exists y \in S. X(x,y) \tag{3}$$

$$\forall x,y,z.((X(x,y) \wedge X(x,z)) \supset y = z) \tag{4}$$

i.e. X is, in fact, a function $\mathbb{N} \to S$.

Definition 17 (Language membership)**.** Let $\mathcal{A} = (A, Q, \delta, q_0, F)$ be a NBA and treat X a binary predicate symbol. We define the formula $X \in \mathcal{L}(\mathcal{A})$ as:

$$X \in A^\omega \ \wedge \ \exists Y \in Q^\omega. \left(\begin{array}{ll} & Y(0,q_0) \\ \wedge & \forall x.\ (Y(x), X(x), Y(\mathsf{s}x)) \in \delta \\ \wedge & \forall x. \exists x' > x.\ Y(x') \in F \end{array} \right) \tag{5}$$

If $\mathcal{A}$ is deterministic and $X \in A^\omega$, we write $q_X(x)$ for the x^{th} state of the run of X on $\mathcal{A}$. Again, this function is primitive recursive, so its graph is provably total in RCA_0. For DBA, we alternatively define $X \in \mathcal{L}(\mathcal{A})$ as:

$$X \in A^\omega \ \wedge \ \forall x. \exists x' > x.\ q_X(x') \in F \tag{6}$$

This 'double definition' will not be problematic for us, since RCA_0 can check if an automaton is deterministic or not and, if so, even prove the equivalence between the two definitions:

Proposition 18. $\mathsf{RCA}_0 \vdash \forall$ *DBA* $\mathcal{A}.((5) \equiv (6))$.

Proof sketch. Let $\mathcal{A} = (A, Q, \delta, q_0, F)$ be a deterministic automaton. For the left-right direction let $Y \in Q^\omega$ be an 'accepting run' of X on $\mathcal{A}$ and show by induction that $Y(x, q_X(x))$. For the right-left direction, we use comprehension to define an 'accepting run' $Y \in Q^\omega$ by:

$$Y(x,q) \quad \equiv \quad q = q_X(x)$$

Clearly the definition of Y is Δ^0_1, and we can show that such Y is a 'correct run' by induction on x. □

Notice that, for a deterministic automaton, the formula for acceptance is *arithmetical* in X, i.e. there are no SO quantifiers. This will be rather important for uniformity in the simulation of cyclic proofs in the next section.

6. An elementary simulation of CA in PA

In this section we will adapt Simpson's approach in [Sim17] for showing that $\mathsf{CA} \subseteq \mathsf{PA}$ into a somewhat more *uniform* result in PA. This essentially constitutes a formalisation of the soundness argument, Prop. 7, in a SO theory conservative over the target fragment of PA. The 'uniformity' we aim for ensures that the possibly non-elementary blowup translating from SO proofs to FO proofs occurs once and for all for a single arithmetical theorem. Only then do we instantiate the theorem (inside PA) by the cyclic proof in question, leading to only elementary blowup.

To give an idea of how the result is obtained, we take advantage of the following aspects of the soundness argument for cyclic proofs:

(1) The Büchi automaton accepting all infinite branches of a cyclic proof is, in fact, deterministic, and so we can express acceptance of a word in this automaton arithmetically.
(2) A branch of invalid sequents and corresponding assignments, as in the proof of Prop. 7, can be uniformly generated from an initial unsatisfying assignment by an arithmetical formula.
(3) Morally, the 'inductive definitions' we are dealing with have finite closure ordinals, and so we need only *arbitrarily often* progressing traces, rather than explicit infinitely progressing traces.

Together, these give us just enough 'wiggle room' to carry out the soundness argument in a sufficiently uniform way.

Throughout this section we will also carefully track how much quantifier complexity is used in theorem statements, since we will later modify this argument to obtain a converse result to Thm. 13.

6.1. **An arithmetically uniform treatment of automata.**

We define an arithmetical corollary of NBA acceptance that is sufficiently strong to formalise the soundness argument for cyclic proofs:

Definition 19. Let $\mathcal{A}$ be a NBA and $X \in A^\omega$. Temporarily let us write:

- $F(x) :=$ "x is a finite run of X on $\mathcal{A}$ ending at a final state".
- $E(z, x, y) :=$ "z extends x to a finite run of X on $\mathcal{A}$ hitting $\geq y$ final states"

We define:

$$\mathrm{ArAcc}(X, \mathcal{A}) \;:=\; \exists x. \left(F(x) \wedge \forall y. \exists z. E(z, x, y)\right)$$

Referring to 3 above, we will need the following result:

Proposition 20. $\mathsf{RCA}_0 \vdash \forall \mathcal{A}.(X \in \mathcal{L}(\mathcal{A}) \supset \mathrm{ArAcc}(X, \mathcal{A}))$.

For intuition, we may consider ω-regular expressions rather than automata, which are of the form $\sum\limits_{i<n} e_i \cdot f_i^\omega$, without loss of generality. The formula ArAcc essentially says that any word accepted by this language has a prefix in some e_i and tail prefixed by f_i^n for each $n \in \mathbb{N}$.

Proof sketch. Fix $\mathcal{A} = (A, Q, \delta, q_0, F)$ and let $X \in \mathcal{L}(\mathcal{A})$. By definition of the latter, let $Y \in Q^\omega$ be an 'accepting run' of X on $\mathcal{A}$, cf. (5). We have that

$$\forall y . \exists z \in Q^* . \text{“}z \text{ is a prefix of } Y \text{ hitting} \geq y \text{ final states in } \mathcal{A}\text{”} \tag{7}$$

by (Σ^0_1-IND) induction on y, appealing to the unboundedness of final states in Y for both the base case and the inductive steps. Now, in the definition of ArAcc, we set x to be the least such z for which (7)[$1/y$] holds. Thus, for any $y \in \mathbb{N}$, we may find an appropriate z making $\mathrm{ArAcc}(X, \mathcal{A})$ true by appealing to (7). The fact that z extends x follows from leastness of x. □

Recall that we may freely add symbols for primitive recursive functions to our language. Let us write $\mathcal{A}^c$ and $\mathcal{A}_1 \sqcup \mathcal{A}_2$ to denote some standard complement and union constructions of automata, say from [KMPM16]. Let $\mathrm{Empty}(\mathcal{A})$ be a standard recursive algorithm, expressed as a Δ_1 formula in $\mathcal{A}$, determining whether $\mathcal{A}$ computes the empty language (say, also from [KMPM16]). We write “$\mathcal{A}_1 \subseteq \mathcal{A}_2$” for $\mathrm{Empty}((\mathcal{A}_1^c \sqcup \mathcal{A}_2)^c)$. We may now present our main 'uniform' result needed to carry out our soundness proof in FO theories.

Theorem 21. $\mathsf{RCA}_0 + \Sigma^0_2$-$\mathsf{IND}$ *proves:*

$$\forall \textit{ DBA } \mathcal{A}_1, \forall \textit{ NBA } \mathcal{A}_2 .\ ((X \in \mathcal{L}(\mathcal{A}_1) \wedge \text{“}\mathcal{A}_1 \subseteq \mathcal{A}_2\text{”}) \supset \mathrm{ArAcc}(X, \mathcal{A}_2)) \tag{8}$$

To prove this, along with Prop. 20 we use the fact that $\mathsf{RCA}_0 + \Sigma^0_2$-$\mathsf{IND}$ is strong enough to formalise some fundamental results of automaton theory for ω-languages:

Lemma 22 (From [KMPM16])**.** *We have the following:*

(1) $\mathsf{RCA}_0 \vdash \forall \mathcal{A}.(\mathrm{Empty}(\mathcal{A}) \equiv \forall X \in A^\omega . X \notin \mathcal{L}(\mathcal{A}))$. *(Lemma 14)*
(2) $\mathsf{RCA}_0 \vdash \forall \mathcal{A}_1, \mathcal{A}_2 .(X \in \mathcal{L}(\mathcal{A}_1 \sqcup \mathcal{A}_2) \equiv (X \in \mathcal{L}(\mathcal{A}_1) \vee X \in \mathcal{L}(\mathcal{A}_2)))$. *(Thm. 13)*
(3) $\mathsf{RCA}_0 + \Sigma^0_2$-$\mathsf{IND} \vdash \forall \mathcal{A}.(X \in \mathcal{L}(\mathcal{A}^c) \equiv X \notin \mathcal{L}(\mathcal{A}))$. *(Thms. 5 and 12)*

Now we can prove the main result of this subsection:

Proof sketch of Thm. 21. Working in RCA_0, fix $\mathcal{A}_1, \mathcal{A}_2$, satisfying the antecedent of (8). We may proceed by Boolean reasoning under Lemma 22: since “$\mathcal{A}_1 \subseteq \mathcal{A}_2$”, i.e. $\mathrm{Empty}((\mathcal{A}_1^c \sqcup \mathcal{A}_2)^c)$, we have that $\forall X.\ X \notin \mathcal{L}((\mathcal{A}_1^c \sqcup \mathcal{A}_2)^c)$, by 1. From 3 we have that $\forall X.\ X \in \mathcal{A}_1^c \sqcup \mathcal{A}_2$, whence from 2 and again 3 we have that $\forall X.(X \in \mathcal{L}(\mathcal{A}_1) \supset X \in \mathcal{L}(\mathcal{A}_2))$. Finally, since $X \in \mathcal{L}(\mathcal{A}_1)$ from the antecedent, we have $X \in \mathcal{L}(\mathcal{A}_2)$, and so we conclude that $\mathrm{ArAcc}(X, \mathcal{A}_2)$ by Prop. 20. □

By the conservativity result, Prop. 16, we have:

Corollary 23. $\mathrm{I}\Sigma_2(X)$ *proves* (8).

6.2. **Formalising the soundness argument for cyclic proofs.** At this point we are able to mostly mimic the formalisation of the soundness argument from [Sim17], although we must further show that a branch of invalid sequents, cf. the proof of Prop. 7, is uniformly describable. Let $\mathbb{N}, \rho \vDash_n \varphi$ be an appropriate Σ_n formula in ρ, φ (as long as φ is also Σ_n) asserting that φ is true in $\mathbb{N}$ under the assignment ρ of its free variables to natural numbers. This is a standard construction, cf. [Bus98, Kay91].

Definition 24 (Uniform description of an invalid branch)**.** Let π be a CA-preproof of a sequent $\Gamma \Rightarrow \Delta$, and let $n \in \mathbb{N}$ be such that all formulae occurring in π are Σ_n. Let ρ_0 be an assignment such that $\mathbb{N}, \rho_0 \vDash_n \bigwedge \Gamma$ but $\mathbb{N}, \rho_0 \nvDash_n \bigvee \Delta$. The branch of π **generated** by ρ_0 is the invalid branch as constructed in the proof of Prop. 7, where at each step that there is a choice of premiss the leftmost one is chosen, and at each step when there is a choice of assignment of a natural number to a free variable the least one is chosen.

We write $\mathrm{Branch}_n(\pi, \rho_0, x, y)$ for the following predicate:

$$\text{“the } x^{\text{th}} \text{ element of the branch generated by } \rho_0 \text{ in } \pi \text{ is } y\text{”}$$

Notice that $\mathrm{Branch}_n(\pi, \rho_0, x, y)$ is recursive in $\vDash_n$, and so is expressible by a $\Delta_1(\vDash_n)$ formula, making it altogether Δ_{n+1} in its arguments. In fact, this is demonstrably the case already in $\mathrm{I}\Sigma_{n+1}$, which can prove that $\mathrm{Branch}_n(\pi, \rho_0, -, -)$ is the graph of a *function*, as shown in the following proposition. We will write $\mathcal{A}_b^\pi$ and $\mathcal{A}_t^\pi$ for the branch and trace automata of a proof π, and $\mathsf{conc}(\pi)$ for its conclusion. When we write $\mathbb{N}, \rho \vDash_n \Gamma \Rightarrow \Delta$ we mean the Δ_{n+1} formula $(\mathbb{N}, \rho \nvDash_n \bigwedge \Gamma) \vee (\mathbb{N}, \rho \vDash_n \bigvee \Delta)$.

Proposition 25. *Let π be a CA-preproof of a sequent $\Gamma \Rightarrow \Delta$, and let $n \in \mathbb{N}$ be such that all formulae occurring in π are Σ_n. $\mathrm{I}\Sigma_{n+1}$ has proofs of size polynomial in n of:*

$$\forall \pi, \rho_0 . (\mathbb{N}, \rho_0 \nvDash_n \mathsf{conc}(\pi) \supset \mathrm{Branch}_n(\pi, \rho_0, -, -) \in \mathcal{L}(\mathcal{A}_b^\pi))$$

Proof idea. Using the definition of $\mathrm{Branch}_n(\pi, \rho_0, -, -)$, the proof follows by a routine induction on the position in the branch. One subtlety is that showing totality requires us to use the *least number principle* for a Σ_n-formula, namely of the form $\exists x . (\mathbb{N}, \rho \vDash_n \varphi(x))$, when dealing with a $\exists$-left or $\forall$-right step. □

On closer inspection, the proof above should go through already in $\mathrm{I}\Delta_{n+1}$, since we can bound y in (6) by x as every state in $\mathcal{A}_b^\pi$ is final. In any case, this does not make a difference to the final bound we obtain.

We are now ready to give our main proof complexity result:

Theorem 26. $\mathsf{CA} = \mathsf{PA}$*, with only elementary difference in proof size.*

Proof sketch. The right-left direction is available already in [Sim17], or alternatively, by Prop. 9.

For the left-right direction, let π be a CA proof containing only Σ_n formulae. Instantiating X in Cor. 23 with a Δ_{n+1} formula yields a $\mathrm{I}\Sigma_{n+2}$ proof of (8), hence from Prop. 25 we arrive at an $\mathrm{I}\Sigma_{n+2}$ proof of:

$$\forall \pi, \rho_0 . \left(\begin{array}{cl} & (\text{“}\mathcal{A}_b^\pi \subseteq \mathcal{A}_t^\pi\text{”} \wedge \mathbb{N}, \rho_0 \nvDash_n \mathsf{conc}(\pi)) \\ \supset & \mathrm{ArAcc}(\mathrm{Branch}(\pi, \rho_0, -, -), \mathcal{A}_t^\pi) \end{array} \right)$$

Now, working inside $\mathrm{I}\Sigma_{n+2}$ and assuming $\mathbb{N}, \rho_0 \nvDash_n \mathsf{conc}(\pi))$, we may derive a contradiction from “$\mathcal{A}_b^\pi \subseteq \mathcal{A}_t^\pi$” by mimicking the proof of Prop. 7. Importantly, if

x witnesses the outer existential of $\mathrm{ArAcc}(X, \mathcal{A})$, we must choose a y bigger than any of the values assigned to terms at the x^{th} sequent of $\mathrm{Branch}(\pi, \rho_0, -, -)$, i.e. at the fixed position where traces of 'arbitrary finite progress' begin. Finally we appeal to the fact that, if π is a correct cyclic proof, there is a proof of "$\mathcal{A}_b^\pi \subseteq \mathcal{A}_t^\pi$", by exhaustive search. (In fact, such a proof will have exponential size in $|\pi|$, since inclusion of Büchi automata is decidable in **PSPACE**.)

Hence, by the reflection property that $(\mathbb{N}, \varnothing \vDash_n \varphi) \equiv \varphi$, we have elementary size proofs of $\mathsf{conc}(\pi)$ in $\mathrm{I}\Sigma_{n+2}$. $\square$

7. $\mathrm{I}\Sigma_{n+1}$ CONTAINS $\mathrm{C}\Sigma_n$

In fact the proof method we developed in the last section allows us to recover a result on logical complexity too. By tracking precisely all the bounds therein, we obtain that $\mathrm{C}\Sigma_n$ is contained in $\mathrm{I}\Sigma_{n+2}$, which is already an improvement to Simpson's result (see Sect. 8 for a comparison). However, we may actually arrive at the optimal result, given Thm. 13, by more carefully analysing the proof methods of [KMPM16], namely an inspection of the proofs of Thms. 5 and 12.

Proposition 27 (Implicit in [KMPM16])**.** *For any NBA $\mathcal{A}$, we have that $\mathsf{RCA}_0 \vdash \forall X \in A^\omega.(X \in \mathcal{L}(\mathcal{A}^c) \equiv X \notin \mathcal{L}(\mathcal{A}))$.*

Notice here that the universal quantification over NBA is *external*, so that the complementation proofs are not necessarily unifoirm. This is not a trivial result, since it relies on a version of Ramsey's theorem, the *additive Ramsey theorem*, which can be proved by induction on the number of 'colours'. Usual forms of Ramsey's theorem are not proved by such an argument, and in fact it is well known that RCA_0 cannot even prove Ramsey's theorem for pairs with only two colours (see, e.g., [Hir14]).

This allows us to 'un-uniformise' the results of the previous section, using Prop. 27 above instead of 3 from Lemma 22, in order to 'trade off' proof complexity for logical complexity.

Theorem 28. *$\mathrm{C}\Sigma_n \subseteq \mathrm{I}\Sigma_{n+1}$, for $n \geq 0$.*

Proof sketch. By the definition of $\mathrm{C}\Sigma_n$, we may assume that all formulae occurring are Σ_n, and hence we may instantiate this value for n in the proof of Thm. 26. The only difference is that, by using Prop. 27 above instead of 3 from Lemma 22 in Thm. 21 and Cor. 23, we obtain the analogous results for RCA_0 and $\mathrm{I}\Sigma_1$ respectively, reducing the induction complexity. Thus the proof of Thm. 26 may be carried out entirely within $\mathrm{I}\Sigma_{n+1}$. $\square$

7.1. **On the proof complexity of $\mathrm{C}\Sigma_n$.** One might be tempted to conclude that the elementary simulation of CA by PA should go through already for $\mathrm{C}\Sigma_n$ by $\mathrm{I}\Sigma_{n+2}$, due to the bounds implicit in the proog of Thm. 26. Furthermore, if we are willing to give up a few more exponentials in complexity, one may even bound the size of $\mathrm{I}\Sigma_1$ proofs arising from Prop. 27 by an appropriate elementary function (though this analysis is beyond the scope of this paper).

However, we must be conscious of the 'robustness' of the definition of $\mathrm{C}\Sigma_n$ proofs in terms of complexity. The one we gave, which essentially requires cyclic proofs to contain only Σ_n-sequents, is more similar to free-cut free $\mathrm{I}\Sigma_n$ proofs than general ones, so it seems unfair to compare $\mathrm{I}\Sigma_n$ and $\mathrm{C}\Sigma_n$ for proof complexity. In fact the following result allows us to define a more natural notion of a $C\Sigma_n$ proof from the point of view of complexity, while inducing the same theory.

First, let us recall some notions from, e.g., [Bro06, BS11]. Any cyclic proof can be written in 'cycle normal form', where any 'backpointer' (e.g., the upper sequents we have marked $\bullet$ until now) points only to a sequent that has occurred below it in the proof. Referring to the terminology of [Bro06, BS11] etc., we say that a sequent is a *bud* in a cyclic proof if it has a backpointer pointing to a lower sequent (called the *companion*).

Proposition 29. *If φ has a* CA *proof whose buds and companions contain only Σ_n formulae then $C\Sigma_n \vdash \varphi$.*

Proof idea. Once again, we apply free-cut elimination, Thm. 4, treating any backpointers as 'initial sequents' for a proof in cycle normal form. □

This again shows the robustness of the definition of $C\Sigma_n$ as a theory, and we would further argue that, from the point of view of proof complexity, the backpointer-condition above constitutes a better notion of $C\Sigma_n$ proof.

At the same time we see that it is not easy to compare the proof complexity of this notion of $C\Sigma_n$ and $I\Sigma_{n+1}$, due to the fact that we have used a free-cut elimination result for the simulations in both directions, inducing a non-elementary blowup in proof size.

8. Conclusions and further remarks

In this work we developed the theory of cyclic arithmetic by studying the logical complexity of its proofs. We showed that inductive and cyclic proofs of the same theorems require similar logical complexity, and obtained tight simulation results in both directions. We further showed that the proof complexity of the two frameworks does not differ substantially, although it remains unclear how to properly measure proof complexity for the fragments $C\Sigma_n$, even if the theory seems well-defined and robust. Many of these issues constitute avenues for further work.

8.1. **Comparison to the proofs of** [BT17b] **and** [Sim17]**.** One reason for our improved quantifier complexity compared to [Sim17], is that Simpson rather relies on *weak König's lemma* (WKL) to obtain an infinite branch. This, *a priori*, increases quantifier complexity of the argument, since WKL is known to be unprovable in RCA_0 even in the presence of Σ^0_2-IND; in fact, it is incomparable to Σ^0_2-IND (see, e.g., [KMPM16]). That said, we point out that the 'bounded-width WKL' (bwWKL) of [KMPM16] in fact seems to suffice to carry out Simpson's proof, and this principle is provable already in $\mathsf{RCA}_0 + \Sigma^0_2$-$\mathsf{IND}$. Applying this strategy to his proof yields only that $C\Sigma_n \subseteq I\Sigma_{n+3}$, since bwWKL is applied to a Π^0_{n+1}-set, but we believe this can be adapted to a $I\Sigma_{n+2}$ bound, by more carefully tracking quantifier complexity. We reiterate that the main improvement here is in terms of the proof complexity of the transformation, cf. Thm. 26.

Berardi and Tatsuta's approach, [BT17b], is rather interesting since it is arguably more 'structural' in nature, relying on proof-level manipulations rather than reflection principles. That said there are still crucial sources of nonconstructivity, namely in an 'arithmetical' version of *Ramsey's theorem* (Thm. 5.2) and the consequent *Podelski-Rybalchenko termination theorem* (Thm. 6.1). Both of these increase quantifier complexity by several levels, and so their approach does not seem to yield comparable logical bounds to this work. However, it does indeed seem from an analysis of their proof that an only elementary blowup occurs when transforming a CA proof to one in PA, matching our Thm. 26.

8.2. **Interpreting ordinary inductive definitions in arithmetic.** In earlier work by Brotherston and Simpson, cyclic proofs were rather considered over a system of FO logic extended by 'ordinary' *Martin-Löf* inductive definitions [ML71], known as FOL_{ID} [Bro06, BS07, BS11]. Berardi and Tatsuta showed in [BT17b] that the cyclic system CLKID^ω for FOL_{ID} is equivalent to the inductive system LKID, when at least arithmetic is present, somewhat generalising Simpson's result [Sim17]. We point out that the two results are arguably equipotent since ordinary Martin-Löf inductive definitions can be *interpreted* in arithmetic, with the necessary properties provable. This is because the closure ordinals for ordinary Martin-Löf inductive definitions are $\leq \omega$, and so a Σ_1 inductive construction of 'approximants' can always determine whether an individual belongs to an inductive predicate or not. Notice that this was crucial for our use of ArAcc over the SO acceptance formula. This is also precisely the role of the 'stage numbers' in [BT17b]; there the fresh inductive predicates P' can be expressed as Δ_0-formulae. In particular, this means that $\mathsf{CLKID}^\omega(+\mathsf{PA})$ is *conservative* over CA. We reiterate that the interest behind the results of [BT17b] is rather the structural nature of the transformations, but this observation also exemplifies why CA is a natural and canonical object of study, cf. [Sim17].

8.3. **Cyclic propositional proof complexity.** One perspective gained from this work comes in the setting of *propositional proof complexity* (see, e.g., [CN10, Kra95]). From the results and methods herein, we may formalise in $\mathrm{C}\Delta_0(X)$, say, a proof of the relativised version of the (finitary) pigeonhole principle, which is known to be unprovable in $\mathrm{I}\Delta_0(X)$ due to lower bounds on propositional proofs of bounded depth [KPW95, PBI93].

At the same time the 'Paris-Wilkie' translation [PW81], which fundamentally links $\mathrm{I}\Delta_0$ to bounded-depth proofs, works locally on a proof, at the level of formulae. Consequently one may still apply the translation to the lines of a $\mathrm{C}\Delta_0$ proof to obtain small 'proof-like' objects containing only formulae of bounded depth, and a cyclic proof structure. One would expect that this corresponds to some strong form of 'extension', since it is known that adding usual extension to bounded systems already yields full 'extended Frege' proofs. However at the same time, some of this power has been devolved to the proof structure rather than simply at the level of the formula, and so could yield insights into how to prove simulations between fragments of Hilbert-Frege systems with extension.

References

[AL17] Bahareh Afshari and Graham E. Leigh. Cut-free completeness for modal mu-calculus. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12, 2017.

[BBC08] James Brotherston, Richard Bornat, and Cristiano Calcagno. Cyclic proofs of program termination in separation logic. In *Proceedings of the 35th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2008, San Francisco, California, USA, January 7-12, 2008*, pages 101–112, 2008.

[BDP11] James Brotherston, Dino Distefano, and Rasmus Lerchedahl Petersen. Automated cyclic entailment proofs in separation logic. In *CADE-23 - 23rd International Conference on Automated Deduction, Wroclaw, Poland, July 31 - August 5, 2011. Proceedings*, pages 131–146, 2011.

[BDS16] David Baelde, Amina Doumane, and Alexis Saurin. Infinitary proof theory: the multiplicative additive case. In *25th EACSL Annual Conference on Computer Science Logic, CSL 2016, August 29 - September 1, 2016, Marseille, France*, pages 42:1–42:17, 2016.

[BGP12] James Brotherston, Nikos Gorogiannis, and Rasmus Lerchedahl Petersen. A generic cyclic theorem prover. In *Programming Languages and Systems - 10th Asian Symposium, APLAS 2012, Kyoto, Japan, December 11-13, 2012. Proceedings*, pages 350–367, 2012.

[Bro05] James Brotherston. Cyclic proofs for first-order logic with inductive definitions. In *Automated Reasoning with Analytic Tableaux and Related Methods, International Conference, TABLEAUX 2005, Koblenz, Germany, September 14-17, 2005, Proceedings*, pages 78–92, 2005.

[Bro06] James Brotherston. *Sequent calculus proof systems for inductive definitions*. PhD thesis, University of Edinburgh, 2006.

[BS07] James Brotherston and Alex Simpson. Complete sequent calculi for induction and infinite descent. In *22nd IEEE Symposium on Logic in Computer Science (LICS 2007), 10-12 July 2007, Wroclaw, Poland, Proceedings*, pages 51–62, 2007.

[BS11] James Brotherston and Alex Simpson. Sequent calculi for induction and infinite descent. *J. Log. Comput.*, 21(6):1177–1216, 2011.

[BT17a] Stefano Berardi and Makoto Tatsuta. Classical system of martin-löf's inductive definitions is not equivalent to cyclic proof system. In *Foundations of Software Science and Computation Structures - 20th International Conference, FOSSACS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings*, pages 301–317, 2017.

[BT17b] Stefano Berardi and Makoto Tatsuta. Equivalence of inductive definitions and cyclic proofs under arithmetic. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12, 2017.

[Bus98] Samuel R Buss. *Handbook of proof theory*, volume 137. Elsevier, 1998.

[CN10] Stephen Cook and Phuong Nguyen. *Logical Foundations of Proof Complexity*. Cambridge University Press, New York, NY, USA, 1st edition, 2010.

[DBHS16] Amina Doumane, David Baelde, Lucca Hirschi, and Alexis Saurin. Towards completeness via proof search in the linear time μ-calculus: The case of büchi inclusions. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 377–386, 2016.

[DHL06] Christian Dax, Martin Hofmann, and Martin Lange. A proof system for the linear time μ-calculus. In *FSTTCS 2006: Foundations of Software Technology and Theoretical Computer Science, 26th International Conference, Kolkata, India, December 13-15, 2006, Proceedings*, pages 273–284, 2006.

[Dou17] Amina Doumane. Constructive completeness for the linear-time μ-calculus. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12, 2017.

[DP17] Anupam Das and Damien Pous. A cut-free cyclic proof system for kleene algebra. In *Automated Reasoning with Analytic Tableaux and Related Methods - 26th International Conference, TABLEAUX 2017, Brasília, Brazil, September 25-28, 2017, Proceedings*, pages 261–277, 2017.

[For14] Jérôme Fortier. *Puissance expressive des preuves circulaires. (Expressive Power of Circular Proofs)*. PhD thesis, Aix-Marseille University, Aix-en-Provence, France, 2014.

[FS13] Jérôme Fortier and Luigi Santocanale. Cuts for circular proofs: semantics and cut-elimination. In *Computer Science Logic 2013 (CSL 2013), September 2-5, 2013, Torino, Italy*, pages 248–262, 2013.

[Hir14] Denis R Hirschfeldt. *Slicing the truth: On the computable and reverse mathematics of combinatorial principles*. World Scientific, 2014.

[Kay91] Richard Kaye. Models of peano arithmetic. 1991.

[KMPM16] Leszek Aleksander Kolodziejczyk, Henryk Michalewski, Pierre Pradic, and Michał Skrzypczak. The logical strength of büchi's decidability theorem. In *25th EACSL Annual Conference on Computer Science Logic, CSL 2016, August 29 - September 1, 2016, Marseille, France*, pages 36:1–36:16, 2016.

[KPW95] Jan Krajíček, Pavel Pudlák, and Alan Woods. An exponential lower bound to the size of bounded depth frege proofs of the pigeonhole principle. *Random Structures & Algorithms*, 7(1):15–39, 1995.

[Kra95] Jan Krajíček. *Bounded arithmetic, propositional logic, and complexity theory*. Cambridge University Press, New York, NY, USA, 1995.

[ML71] Per Martin-Lf. Hauptsatz for the intuitionistic theory of iterated inductive definitions. In J.E. Fenstad, editor, *Proceedings of the Second Scandinavian Logic Symposium*, volume 63 of *Studies in Logic and the Foundations of Mathematics*, pages 179 – 216. Elsevier, 1971.

[NW96] Damian Niwinski and Igor Walukiewicz. Games for the mu-calculus. *Theor. Comput. Sci.*, 163(1&2):99–116, 1996.

[Par72] Charles Parsons. On n-quantifier induction. *The Journal of Symbolic Logic*, 37(3):466–482, 1972.

[PBI93] Toniann Pitassi, Paul Beame, and Russell Impagliazzo. Exponential lower bounds for the pigeonhole principle. *Computational Complexity*, 3:97–140, 1993. 10.1007/BF01200117.

[PW81] J.B. Paris and A.J. Wilkie. Δ_0 sets and induction. *Open Days in Model Theory and Set Theory, W. Guzicki, W. Marek, A. Pelc, and C. Rauszer, eds*, pages 237–248, 1981.

[RB17] Reuben N. S. Rowe and James Brotherston. Automatic cyclic termination proofs for recursive procedures in separation logic. In *Proceedings of the 6th ACM SIGPLAN Conference on Certified Programs and Proofs, CPP 2017, Paris, France, January 16-17, 2017*, pages 53–65, 2017.

[San02] Luigi Santocanale. A calculus of circular proofs and its categorical semantics. In *Foundations of Software Science and Computation Structures, 5th International Conference, FOSSACS 2002, Grenoble, France, April 8-12, 2002, Proceedings*, pages 357–371, 2002.

[Sch77] Kurt Schtte. *Proof Theory*. Grundlehren der mathematischen Wissenschaften 225. Springer Berlin Heidelberg, 1977. Translation of Beweistheorie, 1968.

[SD03] Christoph Sprenger and Mads Dam. On the structure of inductive reasoning: Circular and tree-shaped proofs in the μ-calculus. In *Foundations of Software Science and Computational Structures, 6th International Conference, FOSSACS 2003, Warsaw, Poland, April 7-11, 2003, Proceedings*, pages 425–440, 2003.

[Sim09] Stephen George Simpson. *Subsystems of second order arithmetic*, volume 1. Cambridge University Press, 2009.

[Sim17] Alex Simpson. Cyclic arithmetic is equivalent to peano arithmetic. In *Foundations of Software Science and Computation Structures - 20th International Conference, FOSSACS 2017, Proceedings*, pages 283–300, 2017.