Cyclic Arithmetic Is Equivalent to Peano Arithmetic

Alex Simpson(✉)

Faculty of Mathematics and Physics,
University of Ljubljana, Ljubljana, Slovenia
`Alex.Simpson@fmf.uni-lj.si`

Abstract. *Cyclic proof* provides a style of proof for logics with inductive (and coinductive) definitions, in which proofs are cyclic graphs representing a form of argument by infinite descent. It is easily shown that cyclic proof subsumes proof by (co)induction. So cyclic proof systems are at least as powerful as the corresponding proof systems with explicit (co)induction rules. Whether or not the converse inclusion holds is a non-trivial question. In this paper, we resolve this question in one interesting case. We show that a cyclic formulation of first-order arithmetic is equivalent in power to Peano Arithmetic. The proof involves formalising the meta-theory of cyclic proof in a subsystem of second-order arithmetic.

1 Introduction

Cyclic (or *circular*[1]) proof has been studied by a number of authors, see, e.g., [1,2,4–13,17–19,21,22,24]. It is a style of proof suitable for logics with inductive and coinductive definitions. The main idea is to allow proofs to be given as cyclic graphs, where the cycles capture the looping nature of arguments by induction and coinduction. For this to provide a sound method of reasoning, a global condition needs to be satisfied by the proof structure in order to rule out fallacious circular arguments. The global condition can be seen as defining cyclic proof as a formalisation of the concept of proof by infinite descent.

In [4,5,10,11], Brotherston and the author studied a natural style of cyclic proof for first-order logic extended with ordinary inductive definitions in the style of Martin-Löf [16]. It was shown that cyclic proof subsumes proof by induction. The question of whether the two styles of proof are equivalent in power was left open, but conjectured to have a positive answer. In a paper appearing alongside this one, Berardi and Tatsuta refute this conjecture [3].[2] In general, cyclic proof is

This material is based upon work supported by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF under Award No. FA9550-14-1-0096.

[1] We prefer *cyclic* for two reasons: proofs are given as cyclic graphs; and the phrase "circular proof" is uncomfortably close to "circular argument", which means an argument that goes round in a circle without establishing anything.

[2] Independently of [3], Stratulat announced a number of potential counterexamples to the conjecture in [23].

J. Esparza and A.S. Murawski (Eds.): FOSSACS 2017, LNCS 10203, pp. 283–300, 2017.
DOI: 10.1007/978-3-662-54458-7_17

more powerful than proof by induction (as long as the latter uses only induction principles for the inductive definitions under consideration).

In this paper, we provide a complementary result to the theorem of Berardi and Tatsuta. We study cyclic proof for first-order arithmetic. Our main result is that the resulting *Cyclic Arithmetic* coincides with Peano Arithmetic. Thus, in the context of first-order arithmetic, there is, after all, an equivalence in power between cyclic proof and proof by induction.

From one point of view, the study of cyclic proof is particularly apposite in the context of arithmetic. In Sect. 2, we argue that Cyclic Arithmetic is of intrinsic interest as a natural formalisation of the notion of proof by *infinite descent*. Our theorem thus has potential philosophical value in establishing a non-trivial equivalence between infinite descent and induction. More generally, our result contributes to the broad programme of obtaining a better understanding of potential methods of proof. Cyclic methods, in particular, appear to offer a promising extension to machine-assisted formalised proof [9], particularly for applications in computer-science-oriented logics [1,2,6–8,12,13,17–19,24].

In Sect. 3, we introduce an infinitary proof system, whose ∞*-proofs* are non-well-founded trees. This is sound and complete for the first-order theory of true arithmetic. Then in Sect. 4, we define *cyclic proofs* as the restriction of ∞-proofs to regular trees — those that can be presented as finite (cyclic) graphs. Importantly, the question of whether a finite graph presents a cyclic proof is decidable. We end Sect. 4 with the proof that cyclic proof subsumes proof by induction; i.e., that Cyclic Arithmetic contains Peano Arithmetic.

The results thus far are all direct analogues, in the setting of arithmetic, of results in [11] for general inductive definitions. Nevertheless, we give detailed proofs. In the case of the completeness theorem and of the proof that cyclic proof subsumes induction, we do so because the proofs, in the context of arithmetic, are simpler than the corresponding proofs for inductive definitions. In the case of the soundness and decidability results, the proofs for arithmetic are similar to those in [11]. Nevertheless, we supply the details because they are needed in the proof of our main result, Theorem 6, which states that Cyclic Arithmetic is conservative over Peano Arithmetic.

Theorem 6 is proved in Sect. 5. The proof method is to formalise the soundness argument for ∞-proofs in ACA_0, a subsystem of second-order arithmetic, which has been widely studied in the context of reverse mathematics [20]. The use of second-order logic is essential for formalising soundness because of the infinitary nature of ∞-proofs. The reason for the particular choice of the subsystem ACA_0 is that it is conservative over Peano Arithmetic. Once the soundness of Cyclic Arithmetic has been established in ACA_0, the conservativity of Cyclic Arithmetic then follows from a lemma (Lemma 9) that says that ACA_0 can recognise a cyclic proof when presented with one.

Section 6 is devoted to the proof of Lemma 9. For this, we make use of constructions and results from the theory of Büchi automata, once again formalised in ACA_0. Our presentation builds on the recent work of [15], in which the fundamental complementation result for Büchi automata is studied from the perspective of reverse mathematics.

Finally, in Sect. 7, we discuss directions for further research.

2 Proof by Infinite Descent

The aim of this section is to motivate Cyclic Arithmetic informally as providing a natural style of number-theoretic proof by infinite descent. We begin with a standard example of a proof by infinite descent, establishing that $\sqrt{2}$ is irrational. Since we work in the language of arithmetic, we express this as: there are no natural numbers x_0, x_1 such that $x_0 > 0$ and $x_0^2 = 2x_1^2$.

Suppose, for contradiction, that we have x_0, x_1 such that $x_0 > 0$ and $x_0^2 = 2x_1^2$. It then follows that $x_0 > x_1 > 0$. Since 2 is a prime factor of x_0^2 it must be a prime factor of x_0 itself. So $x_0 = 2x_2$ for some x_2. So $4x_2^2 = 2x_1^2$, hence $x_1^2 = 2x_2^2$.

We have now gone round in a circle back to the start of the proof, but with x_1, x_2 in place of x_0, x_1. By repeating the argument for x_1, x_2 we discover that $x_1 > x_2 > 0$ and $x_2^2 = 2x_3^2$ for some x_3. Continuing, $x_2 > x_3 > 0$ and $x_3^2 = 2x_4^2$; whence $x_3 > x_4 > 0$, etc. So, starting from our initial assumptions that $x_0 > 0$ and $x_0^2 = 2x_1^2$, we produce an infinite strictly descending sequence $x_0 > x_1 > x_2 > x_3 > \dots$ of positive integers. Since no such sequence exists, we have obtained the desired contradiction.

Figure 1 presents this proof as an infinite proof tree of sequents. The steps labelled $(\star)$ and $(\dagger)$ are not intended to be atomic proof steps. Rather $(\star)$ is the main number-theoretic lemma used in the proof, and $(\dagger)$ chains together a few simple steps of arithmetical and logical reasoning (including a cut). The ellipsis at the top right represents the continuation of the argument via an infinite sequence of repetitions of the visible proof pattern, but with the variables changed appropriately at each repetition.

$$\dfrac{\dfrac{}{0<x_0,\ x_0^2 = 2x_1^2 \Rightarrow 0<x_1<x_0 \wedge \exists x_2.\, x_0=2x_2}\,(\star) \qquad \dfrac{\dfrac{\vdots}{0<x_1,\ x_1^2 = 2x_2^2 \Rightarrow \bot}}{\underline{x_1<x_0},\ 0<x_1,\ 4x_2^2 = 2x_1^2 \Rightarrow \bot}}{0<x_0,\ x_0^2 = 2x_1^2 \Rightarrow \bot}\,(\dagger)$$

Fig. 1. Infinite descent proof of the irrationality of $\sqrt{2}$

The proof tree in its entirety is an infinite tree, with one growing up to the right. Going up this branch, the variables $x_0, x_1, x_2, \dots$, once introduced, never change their value. Moreover, we pass through an infinite sequence of underlined statements $x_1 < x_0$, $x_2 < x_1$, $x_3 < x_2$ each appearing as an antecedent (i.e., left-hand formula) in a sequent. It is this fact that makes the argument a valid proof by infinite descent.

$$\dfrac{\dfrac{}{x=0 \Rightarrow A(x,y,y{+}1)} \qquad \dfrac{\dfrac{\overset{\vdots}{\overset{(A)}{\Rightarrow} \exists z.A(x{-}1,1,z)}}{x>0, y=0 \Rightarrow \exists z.A(x,y,z)} \qquad \dfrac{\dfrac{\overset{\vdots}{\overset{(B)}{\Rightarrow} \exists z.A(x,y{-}1,z)} \qquad \overset{\vdots}{\overset{(C)}{\Rightarrow} \exists z.A(x{-}1,y',z)}}{\Rightarrow \exists z, y'.\, A(x,y{-}1,y') \wedge A(x{-}1,y',z)}}{x, y>0 \Rightarrow \exists z.\, A(x,y,z)}}{x>0 \Rightarrow \exists z.A(x,y,z)}}{\Rightarrow \exists z.A(x,y,z)}$$

Fig. 2. Infinite descent proof of the totality of the Ackermann-Péter function.

We present one more example of a proof in a similar style. Let $A(x, y, z)$ be a ternary relation on natural numbers satisfying:

$$\begin{aligned} A(x,y,z) \;\Leftrightarrow\; & (x = 0 \wedge z = y + 1) \\ & \vee\; (x > 0 \wedge y = 0 \wedge A(x-1, 1, z)) \\ & \vee\; (x, y > 0 \wedge \exists w.\, A(x, y-1, w) \wedge A(x-1, w, z)) \end{aligned}$$

This formula defines A to be the graph of the well-known 2-argument Ackermann-Péter function. Using standard techniques of definition, one can encode $A(x, y, z)$ by a Σ^0_1-formula in the language of arithmetic satisfying the equivalence above.

Figure 2 presents a proof by infinite descent of the totality of the Ackermann-Péter function. As before, the individual rules are not atomic steps, but contain arithmetical and logical reasoning, including manipulation of the defining property of $A(x, y, z)$ above. This time, the full infinite proof is built by repeating the basic pattern three times, once each at (A), (B) and (C), *ad infinitum.* In doing this, variables are substituted by the terms specified in each case. Note that infinitely many variables $y, y', y'', \ldots$ appear in the infinite proof.

The proof in Fig. 2 presents a correct argument by infinite descent for the following reason. By the local soundness of all rules in the proof, any x, y providing a counterexample to the concluding sequent will generate an infinite branch together with assignments to all variables such that all sequents along the branch are false. There are now two cases to consider.

– If the infinite branch passes through sequents in positions (A) or (C) infinitely often then the value of the number supplied in the x position of $A(x, y, z)$ is decremented infinitely often even though it remains positive along the path.
– Otherwise, the branch must eventually reach a point after which it avoids (A) and (C). Thus all subsequent repetitions are attained via (B). In this case, the number appearing in the y position, at the point in question, is decremented infinitely often, although it again remains positive.

In either case, we have the desired contradiction.

Thus far, we have been describing proofs by infinite descent as infinite proofs. This is in keeping with the idea that a proof by infinite descent should construct infinite sequences, but it is not compatible with the idea of a proof as a finite representation of an argument. Nonetheless, a very natural restriction on infinite proofs can be imposed to achieve such a finite representation. One simply asks for the infinite proof tree to be *regular*, namely for it to have only finitely many distinct subtrees. Any such regular infinite proof tree can be presented as a finite cyclic graph. For example, consider the version below of the proof from Fig. 1, in which a substitution rule is used to make all subtrees rooted at sequents labelled $(*)$ identical. The full proof tree is thus presented by the finite cyclic graph obtained by identifying the nodes labelled $(*)$.

$$\dfrac{\dfrac{}{0<x_0,\, x_0^2 = 2x_1^2 \Rightarrow 0<x_1<x_0 \wedge \exists x_2.\, x_0 = 2x_2} \qquad \dfrac{\dfrac{\dfrac{\vdots}{0<x_0,\, x_0^2 = 2x_1^2 \overset{(*)}{\Rightarrow} \bot}}{0<x_1,\, x_1^2 = 2x_2^2 \Rightarrow \bot}\,(\mathsf{Sub})}{\underline{x_1<x_0},\, 0<x_1,\, 4x_2^2 = 2x_1^2 \Rightarrow \bot}}{0<x_0,\, x_0^2 = 2x_1^2 \overset{(*)}{\Rightarrow} \bot}$$

It is similarly possible to convert Fig. 2 to a regular infinite proof.

In Sect. 3, we give a precise definition of ∞*-proof* that formalises the notion of infinite proof by infinite descent described informally above, and we show that ∞-proofs are sound and complete for the first-order theory of true arithmetic. Since the notion of proof is infinitary, the completeness result is unsurprising. For example, a similar completeness property is well known to hold for ω*-proofs*, obtained by adding the infinitary ω-rule to (for example) sequent calculus. Nonetheless, there is an important mathematical distinction between ω-proofs and ∞-proofs. The former are given as infinitely-branching well-founded trees. In contrast, ∞-proofs are finitely branching (potentially) non-well-founded trees.

It is an advantage of ∞-proofs that they possess a naturally identifiable subclass of finitely presentable proofs, the *regular* ones, which we introduce as *cyclic proofs* in Sect. 4. Our main result, the coincidence of cyclic proof for arithmetic with Peano Arithmetic, thus establishes that *finitary proof by infinite descent is equivalent to proof by induction.*

3 ∞-proofs

We formulate arithmetic using first-order logic with equality, with signature $(0, s, +, \cdot, <)$, where s is the successor function. The strict order relation $<$ is included as primitive because it is used in the definition of ∞-proof below.

We give a sequent calculus presentation of our proof calculus. For our purposes, a *sequent* $\Gamma \Rightarrow \Delta$ is a pair of finite sets Γ, Δ of formulas. We use standard notational conventions for sequents, such as omitting set delimiters when writing sets, and using comma ‘,’ for union. We write $\Gamma[\theta]$ for the result of applying the same substitution θ (mapping finitely many variables to associated terms) to every formula in Γ. We also write $\Gamma[t_1, \ldots, t_k]$, for terms $t_1, \ldots, t_k$, to mean $\Gamma[t_1/x_1, \ldots, t_k/x_k]$, where $x_1, \ldots, x_k$ are distinct variables left implicit. In such cases, a parallel mention of $\Gamma[u_1, \ldots, u_k]$ always means $\Gamma[u_1/x_1, \ldots, u_k/x_k]$ for the same variables $x_1, \ldots, x_k$.

Our proof system is built from three sets of rules manipulating sequents. Rules for the logical constants (including equality) are presented in Fig. 3. Structural rules, including (Cut), are given in Fig. 4. Finally, basic arithmetic properties are axiomatised, in Fig. 5, by a list of 11 axiom sequents, together with one further inference rule. The axioms and rule of Fig. 5 implement a finitely axiomatised theory of arithmetic corresponding to a natural expansion of Robinson's system Q with $<$ as a primitive relation. As motivated in Sect. 2, our ∞-proofs are infinite trees, where locally each node of the tree is given by an application of one of the rules or axioms in Figs. 3, 4 and 5. We call such a tree a *pre-∞-proof*. In order to qualify as an actual proof, such a tree will need to satisfy a further condition, whose formulation requires the following definition.

$$\dfrac{}{\Gamma \Rightarrow \Delta}\ \Gamma \cap \Delta \neq \emptyset \qquad \dfrac{\Gamma \Rightarrow A, \Delta}{\Gamma, \neg A \Rightarrow \Delta} \qquad \dfrac{\Gamma, A \Rightarrow \Delta}{\Gamma \Rightarrow \neg A, \Delta}$$

$$\dfrac{\Gamma, A, B \Rightarrow \Delta}{\Gamma, A \wedge B \Rightarrow \Delta} \qquad \dfrac{\Gamma \Rightarrow A, \Delta \quad \Gamma \Rightarrow B, \Delta}{\Gamma \Rightarrow A \wedge B, \Delta} \qquad \dfrac{\Gamma, A \Rightarrow \Delta \quad \Gamma, B \Rightarrow \Delta}{\Gamma, A \vee B \Rightarrow \Delta} \qquad \dfrac{\Gamma \Rightarrow A, B, \Delta}{\Gamma \Rightarrow A \vee B, \Delta}$$

$$\dfrac{\Gamma, A[t/x] \Rightarrow \Delta}{\Gamma, \forall x\, A \Rightarrow \Delta} \qquad \dfrac{\Gamma \Rightarrow A[y/x], \Delta}{\Gamma \Rightarrow \forall x\, A, \Delta}\ y \text{ fresh} \qquad \dfrac{\Gamma, A[y/x] \Rightarrow \Delta}{\Gamma, \exists x\, A \Rightarrow \Delta}\ y \text{ fresh} \qquad \dfrac{\Gamma \Rightarrow A[t/x], \Delta}{\Gamma \Rightarrow \exists x\, A, \Delta}$$

$$\dfrac{\Gamma[u_1, u_2] \Rightarrow \Delta[u_1, u_2]}{\Gamma[u_2, u_1],\ u_1 = u_2 \Rightarrow \Delta[u_2, u_1]} \qquad \dfrac{}{\Gamma \Rightarrow t = t,\ \Delta}$$

Fig. 3. Cut-free sequent calculus with equality

$$\dfrac{\Gamma \Rightarrow \Delta}{\Gamma, \Gamma' \Rightarrow \Delta', \Delta}\ (\mathsf{Wk}) \qquad \dfrac{\Gamma,\ A \Rightarrow \Delta \quad \Gamma \Rightarrow A,\ \Delta}{\Gamma \Rightarrow \Delta}\ (\mathsf{Cut}) \qquad \dfrac{\Gamma \Rightarrow \Delta}{\Gamma[\theta] \Rightarrow \Delta[\theta]}\ (\mathsf{Sub})$$

Fig. 4. Weakening, cut and substitution rules

Definition 1 (Precursor, trace, progress). Let $(\Gamma_i \Rightarrow \Delta_i)_{i \geq 0}$ be an infinite branch through a pre-proof. For terms t, t', we say that t' is a *precursor* of t at i if one of the following holds.

$$\begin{array}{rl}
t < u,\, u < v \Rightarrow & t < v \\
t < u,\, u < t \Rightarrow & \\
\Rightarrow & t < u,\, t = u,\, u < t \\
t < 0 \Rightarrow & \\
t < u \Rightarrow & \mathsf{s}(t) < \mathsf{s}(u) \\
\Rightarrow & t < \mathsf{s}(t) \\
t < u,\, u < \mathsf{s}(t) \Rightarrow & \\
\Rightarrow & t + 0 = t \\
\Rightarrow & t + \mathsf{s}(u) = \mathsf{s}(t + u) \\
\Rightarrow & t \cdot 0 = 0 \\
\Rightarrow & t \cdot \mathsf{s}(u) = (t \cdot u) + t
\end{array}$$

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

Fig. 5. Axioms and rules for basic arithmetic

- $\Gamma_i \Rightarrow \Delta_i$ is the conclusion of an application of (Sub), and $t = \theta(t')$ where θ is the substitution used in the rule application.
- $\Gamma_i \Rightarrow \Delta_i$ is the conclusion of an $(u_1 = u_2)$-left rule, and it is possible to write t' and t as $u[u_1, u_2]$ and $u[u_2, u_1]$ respectively, for some term u.
- $\Gamma_i \Rightarrow \Delta_i$ is the conclusion of one of the other rules, and $t' = t$.

We say that a term t *occurs* in a sequent $\Gamma \Rightarrow \Delta$ if it appears within some formula in Γ, Δ (possibly as a subterm of another term). 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, for every $i \geq N$, the term t_i occurs in $\Gamma_i \Rightarrow \Delta_i$, and also one of the following holds.

- Either t_{i+1} is a precursor of t_i at i,
- or there exists $(t_{i+1} < t) \in \Gamma_{i+1}$ such that t is a precursor of t_i at i.

When the latter case holds, we say that the trace *progresses* at $i + 1$.

Definition 2 (∞-proof). An ∞-*proof* is a pre-∞-proof that satisfies the following *trace condition.*

> Along every infinite branch $(\Gamma_i \Rightarrow \Delta_i)_i$ there exist $N \geq 0$ and a trace $(t_i)_{i \geq N}$ that progresses at infinitely many i.

Modulo the expansion of the depicted rules into combinations of primitive rules from Figs. 3, 4 and 5, the proofs in Figs. 1 and 2 are both ∞-proofs, for the reasons explained in Sect. 2. (E.g., the required trace in Fig. 1 is $x_0, x_1, x_1, x_2, x_2, \ldots$.)

Semantically, we will be interested only in the standard interpretation in the natural numbers $\mathbb{N}$. We write $\mathbb{N} \models^\rho A$ to say that formula A is true in $\mathbb{N}$ under an environment ρ that interprets the free variables of A as natural numbers. We write $\mathbb{N} \models^{\rho} \Gamma \Rightarrow \Delta$ to mean: if $\mathbb{N} \models^{\rho} A$ for all $A \in \Gamma$ then there exists $B \in \Delta$ such that $\mathbb{N} \models^{\rho} B$. We define $\mathbb{N} \models \Gamma \Rightarrow \Delta$ to mean: $\mathbb{N} \models^{\rho} \Gamma \Rightarrow \Delta$ for all $\mathbb{N}$-environments ρ.

Theorem 3 (Soundness for ∞-proofs). *If $\Gamma \Rightarrow \Delta$ has an ∞-proof then $\mathbb{N} \models \Gamma \Rightarrow \Delta$.*

Proof. We suppose, for contradiction, that we have an ∞-proof of $\Gamma_0 \Rightarrow \Delta_0$, but that $\mathbb{N} \not\models^{\rho_0} \Gamma_0 \Rightarrow \Delta_0$.

We first construct an infinite branch $(\Gamma_i \Rightarrow \Delta_i)_i$, together with an associated sequence $(\rho_i)_i$ of environments, such that $\mathbb{N} \not\models^{\rho_i} \Gamma_i \Rightarrow \Delta_i$ for all i. To do this, $\Gamma_{i+1} \Rightarrow \Delta_{i+1}$ and ρ_{i+1} are constructed from $\Gamma_i \Rightarrow \Delta_i$ and ρ_i as follows. Since $\mathbb{N} \not\models^{\rho_i} \Gamma_i \Rightarrow \Delta_i$, the sequent $\Gamma_i \Rightarrow \Delta_i$ must be the conclusion of an inference rule. If this rule is an instance of (Sub) then define $\rho_{i+1} = \rho_i \circ \theta$, where θ is the substitution in the rule. Otherwise define $\rho_{i+1} = \rho_i$. By the soundness of inference rules, at least one premise of the rule is a sequent $\Gamma' \Rightarrow \Delta'$ for which $\mathbb{N} \not\models^{\rho_{i+1}} \Gamma' \Rightarrow \Delta'$. We define $\Gamma_{i+1} \Rightarrow \Delta_{i+1}$ to be a chosen such premise.

By the trace condition, the infinite branch has a trace $(t_i)_{i \geq N}$ that progresses infinitely often. Consider the associated sequence of numbers $(t_i^{\rho_i})_{i \geq N}$. Since $\mathbb{N} \not\models^{\rho_i} \Gamma_i \Rightarrow \Delta_i$, we have that $\mathbb{N} \models^{\rho_i} A$ for every $A \in \Gamma_i$. So, using the definitions of precursor and of ρ_{i+1}, if t_{i+1} is a precursor of t then ${t_{i+1}}^{\rho_{i+1}} = t^{\rho_i}$. Therefore,

- $t_{i+1}^{\rho_{i+1}} = t_i^{\rho_i}$, if t_{i+1} is a precursor of t_i; and
- $t_{i+1}^{\rho_{i+1}} < t_i^{\rho_i}$, if $(t_{i+1} < t) \in \Gamma_{i+1}$ and t is a precursor of t_i.

By the trace condition, the second case applies infinitely often. Thus $(t_i^{\rho_i})_{i \geq N}$ is an infinite non-increasing sequence of natural numbers that decreases infinitely often, which gives the desired contradiction. □

Due to their infinitary nature, ∞-proofs are complete. Indeed, completeness holds even for proofs that contain no instances of (Wk) and (Sub), and in which (Cut) occurs only in cases in which the cut formula A is atomic. We call such proofs *atomic cut* ∞-proofs.

Theorem 4 (Atomic-cut completeness for ∞-proofs). *If $\mathbb{N} \models \Gamma \Rightarrow \Delta$ then there exists an atomic-cut ∞-proof of $\Gamma \Rightarrow \Delta$.*

Proof. The main observation required is that the sequent calculus ω-rule:

$$\frac{\Gamma[0] \Rightarrow \Delta[0] \quad \Gamma[s(0)] \Rightarrow \Delta[s(0)] \quad \Gamma[s(s(0))] \Rightarrow \Delta[s(s(0))] \quad \dots \quad \dots \quad \dots}{\Gamma[t] \Rightarrow \Delta[t]}$$

is simulated by the ∞-proof below (we combine multiple rules into single steps).

$$\dfrac{\dfrac{\varGamma[0] \Rightarrow \varDelta[0]}{t=0, \varGamma[t] \Rightarrow \varDelta[t]} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\varGamma[s(0)] \Rightarrow \varDelta[s(0)]}{x_1=0, \varGamma[s(x_1)] \Rightarrow \varDelta[s(x_1)]} \qquad \dfrac{\dfrac{\dfrac{\varGamma[s(s(0))] \Rightarrow \varDelta[s(s(0))] \qquad \begin{array}{c}\vdots\\ x_2>0, \varGamma[s(s(x_2))] \Rightarrow \varDelta[s(s(x_2))]\end{array}}{\underline{x_2<x_1}\,, \varGamma[s(s(x_2))] \Rightarrow \varDelta[s(s(x_2))]}}{x_1=s(x_2), \varGamma[s(x_1)] \Rightarrow \varDelta[s(x_1)]}}{x_1>0, \varGamma[s(x_1)] \Rightarrow \varDelta[s(x_1)]}}{\underline{x_1<t}\,, \varGamma[s(x_1)] \Rightarrow \varDelta[s(x_1)]}}{t=s(x_1), \varGamma[t] \Rightarrow \varDelta[t]}}{t>0, \varGamma[t] \Rightarrow \varDelta[t]}}{\varGamma[t] \Rightarrow \varDelta[t]}$$

To apply the above, we plug in an ∞-proof for each premise $\varGamma[\underline{n}] \Rightarrow \varDelta[\underline{n}]$. The resulting pre-$\infty$-proof has just one additional infinite branch, the rightmost branch. Along this branch, the sequence $t, t, t, x_1, x_1, x_1, x_2, x_2, x_2, \ldots$ is an infinitely progressing trace. (The progress points are underlined.) $\square$

4 Cyclic Arithmetic

A (possibly infinite) tree is said to be *regular* if it has only finitely many distinct subtrees. Equivalently, a tree is regular if it can be defined as an unfolding of a finite directed (possibly cyclic) graph.

We shall be interested in regular ∞-proofs; that is, in ∞-proofs whose underlying pre-∞-proofs are regular trees. For this, we consider a regular pre-∞-proof to be presented by a finite graph of the following form. Each vertex v of the graph is labelled with an instance $\mathsf{rule}(v)$ of one of the rules in Figs. 3, 4 and 5. We write $\mathsf{conc}(v)$ for the sequent that is the conclusion of the rule instance, and $\mathsf{prem}_i(v)$ for the sequent that is the i-th premise. (Axioms are considered as rules with 0 premises.) When a vertex v has label $\mathsf{rule}(v)$ with n premises, the graph must contain exactly n edges $e_v^1, \ldots, e_v^n$ with source v. Moreover, for each $i = 1, \ldots, n$, it must hold that $\mathsf{prem}_i(v) = \mathsf{conc}(v_i)$, where v_i is the target vertex of e_v^i. Finally, there is a distinguished vertex ε, which represents the conclusion of the pre-∞-proof; i.e., $\mathsf{conc}(\varepsilon)$ is the conclusion sequent.

It is straightforward to see how each such finite graph presentation unfolds to a regular pre-∞-proof, and conversely how each regular pre-∞-proof can be given such a presentation; see [11] for a detailed treatment.

We next establish the decidability of the global trace condition for regular pre-∞-proofs, for which we follow analogous arguments in [11,21]. Although the relatively simple construction does not provide a practical decision procedure, it has the advantage of facilitating the proofs in Sect. 6 below.

Recall that a *(nondeterministic) Büchi automaton* over an alphabet $\varSigma$ is just a nondeterministic finite automaton over $\varSigma$ with an acceptance condition defined for *infinite* words as follows. An *accepting run* of a Büchi automaton B is an infinite sequence of consecutive transitions, starting from an initial state, that passes through an accepting state infinitely often. An infinite word $X \in \Sigma^\omega$ is *accepted* by B if there exists an accepting run labelled by X.

Theorem 5. *It is decidable whether a regular pre-∞-proof, presented as a finite directed graph, is an ∞-proof.*

Proof. Let Σ be the alphabet whose symbols are the edges in the finite graph G that presents the proof. We construct a Büchi automaton B_t over Σ that recognises those infinite paths through G that possess an infinitely progressing trace. The states of B_t are of the form:

– (v) where v is a vertex in G.
– (v, t) where v is a vertex in G and t is a term in $\mathsf{conc}(v)$.
– $(v, t, \checkmark)$ where v is a vertex in G and t is a term in $\mathsf{conc}(v)$.

The transitions are of the form:

– $(v)\xrightarrow{e^i_v}(v')$ and $(v)\xrightarrow{e^i_v}(v', t')$ whenever v' is the target of e^i_v.
– $(v, t)\xrightarrow{e^i_v}(v', t')$ and $(v, t, \checkmark)\xrightarrow{e^i_v}(v', t')$ whenever v' is the target of e^i_v and t' is a precursor of t for $\mathsf{rule}(v)$.
– $(v, t)\xrightarrow{e^i_v}(v', t', \checkmark)$ and $(v, t, \checkmark)\xrightarrow{e^i_v}(v', t', \checkmark)$ whenever v' is the target of e^i_v and $(t' < t'')$ is an antecedent of $\mathsf{conc}(v')$ for some precursor t'' of t for $\mathsf{rule}(v)$.

The accepting states are those with a $\checkmark$ component. The start state is ε. It is clear that an ω-word is accepted if and only if it defines an infinite path through the pre-∞-proof that possess an infinitely progressing trace.

We also need a Büchi automaton B_p that recognises the language of all infinite paths through the pre-∞-proof. The construction of this is trivial, hence omitted.

The pre-∞-proof is an ∞-proof if and only if there is an inclusion of ω-languages $\mathcal{L}(B_p) \subseteq \mathcal{L}(B_t)$. Such inclusions are decidable. $\square$

Since regular ∞-proofs are presented as finite cyclic graphs, we call such proofs *cyclic proofs*, following the terminology of [11]. Similarly, we call the first-order theory consisting of all sentences A such that the sequent $\Rightarrow A$ has a cyclic proof *Cyclic Arithmetic* (CA).

The main result of this paper is that Cyclic Arithmetic coincides with Peano Arithmetic (PA). To ease the comparison, we assume that PA is also formulated in the language $(0, s, +, \cdot, <)$.

Theorem 6 (Coincidence theorem). $\mathsf{CA} = \mathsf{PA}$.

The proposition below establishes the easier inclusion of Theorem 6. The converse inclusion is left to Sect. 5.

Proposition 7. $\mathsf{PA} \subseteq \mathsf{CA}$.

Proof. The axioms and rule of Figs. 3, 4 and 5 capture all of PA except for the induction schema. Moreover, since we have the cut rule, the cyclic-provable sentences are closed under logical consequence. We thus need only show that every instance of induction has a cyclic proof. Such a proof is given by (we identify the $(*)$ nodes):

$$\dfrac{\dfrac{}{x_0 = 0, A[0] \Rightarrow A[x_0]} \qquad \dfrac{\dfrac{\dfrac{\dfrac{A[0], \forall y.(A[y] \to A[s(y)]) \overset{(*)}{\Rightarrow} A[x_0]}{A[0], \forall y.(A[y] \to A[s(y)]) \Rightarrow A[x_1]}\,(\mathsf{Sub})}{\underline{x_1 < x_0}, A[0], \forall y.(A[y] \to A[s(y)]) \Rightarrow A[s(x_1)]}}{x_0 = s(x_1), A[0], \forall y.(A[y] \to A[s(y)]) \Rightarrow A[x_0]}}{x_0 > 0, A[0], \forall y.(A[y] \to A[s(y)]) \Rightarrow A[x_0]}}{A[0], \forall y.(A[y] \to A[s(y)]) \overset{(*)}{\Rightarrow} A[x_0]}$$

The infinite trace is $(x_0\, x_0\, x_0\, x_1\, x_1\, x_1)^\omega$. This indeed progresses infinitely often, at the point underlined in the proof. □

5 Conservativity of CA over PA

The goal of this section is to prove that $\mathsf{CA} \subseteq \mathsf{PA}$. The first step is to prove the soundness of ∞-proofs in ACA_0, a well-known subsystem of second-order arithmetic, which enjoys the property of being conservative over PA. The use of a second-order language allows the formalisation of concepts associated with infinite trees and infinite paths through them, which is necessary for reasoning about ∞-proofs.

Recall (see [20] for a detailed exposition) that the language of second-order arithmetic extends our first-order language with: set variables $X, Y, Z, \ldots$; with quantification over set variables; and with a new atomic formula $t \in X$, where t is a first-order term and X a set variable. A formula is said to be *arithmetical* if it does not contain any set quantifiers (it may contain free set variables). The theory ACA_0 contains the usual first-order axioms of arithmetic, the expected quantifier rules for set variables, and the two principles below.

– The *induction axiom*:

$$\forall X.\ 0 \in X \ \wedge\ (\forall x.\, x \in X \to s(x) \in X) \ \to\ \forall x.\, x \in X.$$

– The *arithmetical comprehension schema*:

$$\exists X.\, \forall x.\, (x \in X \ \leftrightarrow\ \phi),$$

where ϕ ranges over arithmetical formulas in which the set variable X does not occur free.

We assume reasonable encodings of ordered pairs, and sequences as numbers, in which each ordered pair and sequence has a unique encoding. A set X of natural numbers *encodes a tree* if:

- every $x \in X$ encodes a sequence $x_1 \dots x_k$ for some $k \geq 0$;
- if (an encoding of) $x_1 \dots x_k x_{k+1} \in X$ then also $x_1 \dots x_k \in X$; and
- $\varepsilon \in X$ (where ε encodes the empty sequence).

A (partial) function is encoded as a set X of (codes of) ordered pairs satisfying: if $(x, y) \in X$ and $(x, z) \in X$ then $y = z$. The *domain* of X is $\{x \mid \exists y.\ (x, y) \in X\}$. We say that a set X *encodes a labelled tree* if it encodes a function whose domain is a tree. In this case we call the elements of the domain of X the *nodes* of X, and we call the result of applying the function to a node x the *label* of x.

We shall encode ∞-proofs as trees labelled with Gödel numbers of instances of rules from Figs. 3, 4 and 5. For this, we assume a reasonable Gödel numbering of terms, formulas, sequents and rule instances.

Let X encode a labelled tree. We say that X *encodes a pre-∞-proof* if:

- for every $(x, y) \in X$, we have that y encodes a valid rule instance $\mathsf{rule}(x)$;
- if $\mathsf{rule}(x_1 \dots x_k)$ has n premises then the n sequences $x_1 \dots x_k 1, \dots, x_1 \dots x_k n$ are all nodes in X, and no other sequence $x_1 \dots x_k i$ is a node in X; and
- if $x_1 \dots x_k i$ is a node in X then $\mathsf{prem}_i(x_1 \dots x_k) = \mathsf{conc}(x_1 \dots x_k i)$, where we write $\mathsf{prem}_i(x)$ for the i-th premise of $\mathsf{rule}(x)$ and $\mathsf{conc}(x)$ for the conclusion.

An *infinite branch* through a tree X is given by a function Y with domain $\mathbb{N}$, such that $Y(0) = \varepsilon$ and, for every x, it holds that $Y(s(x))$ is a child in X of $Y(x)$. A set Z encodes a sequence $(t_i)_{i \geq N}$ of terms if it is a partial function with domain $\{x \mid x \geq N\}$ and, for every $x \geq N$, it holds that $Z(x)$ is the Gödel number of a term. If X encodes a pre-∞-proof, Y is an infinite branch through X, and Z is a sequence of terms then Definition 1 can be directly translated into the language of second-order arithmetic to define (arithmetical) formulas that express each of the properties:

- Z is a trace along Y in X;
- the trace Z progresses at i in the branch Y of X;
- the trace Z progresses infinitely often in the branch Y of X.

(Note that X needs to appear explicitly in the above formulas because the labelling containing the information about which inference rules are applied is present only in X.) One can thus directly formalize Definiton 2 to obtain a (non-arithmetical) formula expressing:

- X is an ∞-proof.

We wish to prove the soundness of ∞-proofs in ACA_0. Thus, we would like to show that, when we have an ∞-proof of a sequent, that sequent is true (under any interpretation of its free variables). However, the above statement cannot be formulated in ACA_0, because truth is a non-arithmetical property of first-order-arithmetic formulas. We circumvent this by bounding the logical complexity of formulas. For every $n \geq 0$, there is a first-order-arithmetic formula $\mathsf{Tr}_n(x, y)$ which holds if and only if: x is the Gödel number $\ulcorner A \urcorner$ of a Σ^0_n-formula A, and y is the encoding $\ulcorner \rho \urcorner$ an environment ρ, assigning numbers to all free variables in A, such that $\mathbb{N} \models^\rho A$. (See [14, Chap. 9] for a careful construction of such a formula.) If A is a Σ^0_n-sentence then PA proves the truth-reflection schema:

$$\mathsf{Tr}_n(\ulcorner A \urcorner, \ulcorner \emptyset \urcorner) \;\leftrightarrow\; A. \tag{1}$$

Lemma 8 (Formalised soundness). *For every $n \geq 0$, ACA_0 proves:* "for all X, if X is an ∞-proof, containing formulas of complexity at most Σ^0_n, then, for every assignment ρ of numbers to all free variables in the conclusion sequent $\Gamma \Rightarrow \Delta$ of X, the formula $\bigwedge \Gamma \rightarrow \bigvee \Delta$ is Σ^0_{n+1}-true under ρ*".*

In the statement of the lemma, we use quoted sans-serif to emphasise the scope of the formal statement proved in ACA_0. Notice that this formal statement is really a statement about encodings of ∞-proofs and Gödel numbers of sequents and formulas. We have stated it informally for readability. Note also that the use of the Σ^0_{n+1}-truth predicate is appropriate because $\bigwedge \Gamma \rightarrow \bigvee \Delta$ is a Δ^0_{n+1}-formula. For convenience, we henceforth write the sequent $\Gamma \Rightarrow \Delta$ in place of the formula $\bigwedge \Gamma \rightarrow \bigvee \Delta$ in order to talk directly about truth and falsity of sequents.

Proof. We formalise the proof of Theorem 3 in ACA_0. So again suppose we have an ∞-proof X of $\Gamma_0 \Rightarrow \Delta_0$, but that $\Gamma_0 \Rightarrow \Delta_0$ is Σ^0_{n+1}-false under ρ_0.

The main point that requires elaboration is the construction, in ACA_0, of the infinite path $(\Gamma_i \Rightarrow \Delta_i)_i$ in combination with the associated sequence $(\rho_i)_i$ of environments. For this, we assign, to every node x in X, an environment ρ_x:

- $\rho_\varepsilon = \rho_0$.
- $\rho_{x_1 \ldots x_k x_{k+1}} = \begin{cases} \rho_{x_1 \ldots x_k} & \text{if } \mathsf{rule}(x_1 \ldots x_k) \text{ is not an instance of } (\mathsf{Sub}) \\ \rho_{x_1 \ldots x_k} \circ \theta & \text{if } \mathsf{rule}(x_1 \ldots x_k) \text{ is a } (\mathsf{Sub}) \text{ with substitution } \theta \end{cases}$

Let Y be the set (whose definition, for unbounded n, requires full arithmetical comprehension):

$$\{x \in X \mid \text{ for all prefixes } x' \text{ of } x, \text{ the sequent } \mathsf{conc}(x') \text{ is } \Sigma^0_{n+1}\text{-false under } \rho_{x'}\}.$$

Then Y is a finitely branching tree. Moreover, by the soundness of inference rules, every $x \in Y$ has a child node $x' \in Y$. Thus Y is infinite. Hence, by König's lemma (which is a theorem of ACA_0 [20]), there exists an infinite branch Z in Y. Then $i \mapsto \mathsf{conc}(Z(i))$ defines $(\Gamma_i \Rightarrow \Delta_i)_i$ and $i \mapsto \rho_{Z(i)}$ defines $(\rho_i)_i$.

The remainder of the proof, which argues that an infinitely progressing trace along Z contradicts the falsity of the sequents in Z, goes through exactly as in the proof of Theorem 3. □

We next consider cyclic proofs. Any such is presented by a finite graph of the kind introduced at the start of Sect. 4. We assume a sensible encoding of such graphs as natural numbers, and we write $\ulcorner G \urcorner$ for the number representing G. Given $\ulcorner G \urcorner$, we define in ACA_0 the pre-∞-proof generated by G as the set $\mathsf{unfold}(\ulcorner G \urcorner)$ defined as:

$$\{(x_1 \ldots x_k,\, y) \mid \text{there is a path } \varepsilon\, e^{x_1}\, v_1\, e^{x_2}\, v_2 \ldots e^{x_k}\, v_k \text{ in } G, \text{ and } y = \mathsf{rule}(v_k)\}.$$

This indeed defines a set since the defining formula is arithmetical.

Lemma 9. *If G is a finite graph presentation of a regular ∞-proof then ACA_0 proves "$\mathsf{unfold}(G)$ is an ∞-proof".*

Here, and henceforth, we write $\mathsf{unfold}(G)$ rather than $\mathsf{unfold}(\ulcorner G \urcorner)$, since the latter is the default interpretation of the former. Whenever we refer to a finite combinatorial object within ACA_0, we always do so via its numerical encoding.

We postpone the proof of Lemma 9 to Sect. 6 below. Modulo this one pending proof, we now have all the ingredients to complete the proof of Theorem 6.

Proposition 10. $\mathsf{CA} \subseteq \mathsf{PA}$.

Proof. Suppose G is a finite graph presentation of a regular ∞-proof with conclusion sequent $\Rightarrow A$, where A is a sentence. Let n be such that every formula in G is at most Σ^0_n. By Lemma 9, ACA_0 proves "$\mathsf{unfold}(G)$ is an ∞-proof". Whence, by Lemma 8, ACA_0 proves "A is Σ^0_{n+1}-true". Therefore, by an application of the reflection property (1) of the Σ^0_{n+1}-truth predicate, ACA_0 proves A. Since ACA_0 is conservative over PA, it follows that PA proves A. □

Propositions 7 and 10 together establish Theorem 6.

6 Büchi Automata in ACA_0

It remains to prove Lemma 9. Our method of proof is to apply the theory of Büchi automata as formalised in ACA_0. Since a Büchi automaton B over a finite alphabet Σ is a finite combinatorial object, it can be encoded as a natural number $\ulcorner B \urcorner$. We thus formalise properties of Büchi automata as properties of their codes, but we continue with our policy of omitting explicit reference to codes. In ACA_0, an infinite word is coded as a set X defining an infinite sequence of elements of Σ. The property:

$$B \text{ accepts the infinite word } X$$

is directly expressible by a (non-arithmetical) formula in second-order arithmetic. Using this, we formalise the following properties of Büchi automata in ACA_0.

Proposition 11 (Formalised Büchi intersection). *There is a computable binary function $\sqcap$ such that ACA_0 proves: "for all Büchi automata B_1, B_2, it holds that $B_1 \sqcap B_2$ is a Büchi automaton satisfying:*

$$\text{for every infinite } \Sigma\text{-word } X,\ B_1 \sqcap B_2 \text{ accepts } X \ \leftrightarrow\ B_1 \text{ and } B_2 \text{ both accept } X''.$$

Proposition 12 (Formalised Büchi complementation). *There is a computable unary function* $(\cdot)^c$ *such that* ACA_0 *proves:* "for every Büchi automaton B it holds that B^c is a Büchi automaton satisfying:

$$\text{for every infinite } \Sigma\text{-word } X,\ B^c \text{ accepts } X \ \leftrightarrow\ B \text{ does not accept } X''.$$

The above propositions hold because the standard proofs are directly formalisable in ACA_0. Complementation is the more interesting of the two cases since its proof involves nontrivial infinite combinatorics. For example, the crucial step in Büchi's original proof invokes the infinite Ramsey Theorem for pairs (see, e.g., [25]). This does not present a problem for the formalisation because the (general) infinite Ramsey Theorem holds in ACA_0 (see, e.g., [20]). A full analysis can be found in the recent paper [15], where it is shown that Proposition 12 is equivalent to the schema Σ^0_2-IND of Σ^0_2-induction, relative to the weak subsystem of second-order arithmetic RCA_0. In particular, this means that Proposition 12 is provable in $\mathsf{RCA}_0 + \Sigma^0_2$-$\mathsf{IND}$, which is a subsystem of ACA_0.

If B is a Büchi automaton, we write $\rightarrow^*_B$ for the *reachability* relation on states of B (the transitive-reflexive closure of the label-erased transition relation $\rightarrow_B$). The ternary relation "B is a Büchi automaton and $y \rightarrow^*_B z$" is definable by a formula in second-order arithmetic with free first-order variables x, y, z where the parameter x supplies B via its code.

Proposition 13 (Formalised non-emptiness criterion). ACA_0 *proves:* "for every Büchi automaton B, there exists $X \in \Sigma^\omega$ accepted by B if and only if there exist states q_0, q_1 of B such that q_0 is initial, q_1 is accepting and $q_0 \rightarrow^*_B q_1 \rightarrow^*_B q_1$*".*

We omit the proof, which is once again a straightforward formalisation of the (this time easy) standard argument.

Lemma 14. *If B is a Büchi automaton recognising the empty language then* ACA_0 *proves:* "for every infinite Σ-word X, the automaton B does not accept X*".*

Proof. Suppose B recognises the empty language. By the standard non-formalised version of Proposition 13, there do not exist states q_0, q_1 of B such that q_0 is initial, q_1 is accepting and $q_0 \rightarrow^*_B q_1 \rightarrow^*_B q_1$. Since B is finite, the relation $\rightarrow^*_B$ can be encoded as a natural number $\ulcorner\rightarrow^*_B\urcorner$. Easily, ACA_0 proves: "$\ulcorner\rightarrow^*_B\urcorner$ encodes a transitive, reflexive relation R that contains $\rightarrow_B$, and there do not exist states q_0, q_1 of B such that q_0 is initial, q_1 is accepting and $q_0 \mathrel{R} q_1 \mathrel{R} q_1$". Since $\rightarrow^*_B$ is the smallest transitive-reflexive relation containing $\rightarrow_B$, it follows that ACA_0 proves: "there do not exist states q_0, q_1 of B such that q_0 is initial, q_1 is accepting and $q_0 \rightarrow^*_B q_1 \rightarrow^*_B q_1$". Hence, by Proposition 13, ACA_0 proves: "for every infinite Σ-word X, the automaton B does not accept X". □

Proof (of Lemma 9). Let G be a finite graph presentation of a regular pre-∞-proof. Consider the Büchi automata B_t and B_p defined in the proof of Theorem 5. By the definitions of B_t and B_p, the following statements are provable in ACA_0.

– "For all X, B_t accepts X if and only if X is a path through G that possesses an infinitely progressing trace."

– “For all X, B_p accepts X if and only if X is an infinite path through G.”

Hence, by Propositions 11 and 12, ACA_0 proves:

$$\text{“unfold}(G) \text{ is an } \infty\text{-proof iff there is no } X \text{ such that } {B_t}^c \sqcap B_p \text{ accepts } X\text{”}. \quad (2)$$

Now, assume G presents an ∞-proof. We have, as in the proof of Theorem 5, $\mathcal{L}(B_t) \subseteq \mathcal{L}(B_p)$, hence the language of the Büchi automaton ${B_t}^c \sqcap B_p$ is empty. Hence, by Lemma 14, ACA_0 proves: “there is no X such that ${B_t}^c \sqcap B_p$ accepts X”. It thus follows from (2) that ACA_0 proves: “unfold(G) is an ∞-proof”, as required. □

7 Further Work

The following are natural possible continuations of the work in this paper.

(i) Give a syntactic rewrite-based proof eliminating non-atomic cuts from ∞-proofs. (This is done in [2,13] for different notions of cyclic proof.)

(ii) Give an explicit syntactic translation from cyclic proofs to proofs in PA. Is there an essential blow-up in size? (Because of Solovay's speed-up theorem for ACA_0 over PA, the indirect translation implicit in our proof has a potential non-elementary blow-up.)

(iii) Understand the power of cyclic proof in the context of weaker fragments of arithmetic. For example, what is the power of cyclic proof if restricted to Σ^0_1-sequents? The example in Fig. 2 shows that the resulting theory is stronger than $\mathsf{I}\Sigma_1$.

(iv) If the rules and axioms in Figs. 3, 4 and 5 are changed to their intuitionistic versions is the resulting notion of cyclic proof conservative over Heyting Arithmetic?

(v) Understand in general terms when cyclic proof is conservative over proof by induction (as in this paper), and when it is not (as in [3]).

References

1. Baelde, D.: On the proof theory of regular fixed points. In: Giese, M., Waaler, A. (eds.) TABLEAUX 2009. LNCS (LNAI), vol. 5607, pp. 93–107. Springer, Heidelberg (2009). doi:10.1007/978-3-642-02716-1_8
2. Baelde, D., Doumane, A., Saurin, A.: Infinitary proof theory: the multiplicative additive case. In: Talbot, J.-M., Regnier, L. (eds.) 25th EACSL Annual Conference on Computer Science Logic (CSL 2016). Leibniz International Proceedings in Informatics (LIPIcs), Dagstuhl, Germany, vol. 62, pp. 42:1–42:17. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2016)
3. Berardi, S., Tatsuta, M.: Classical system of Martin-Löf's inductive definitions is not equivalent to cyclic proof system. In: Esparza, J., Murawski, A.S. (Eds.) FOSSACS 2017. LNCS, vol. 10203, pp. 301–317. Springer, Heidelberg (2017)
4. Brotherston, J.: Cyclic proofs for first-order logic with inductive definitions. In: Beckert, B. (ed.) TABLEAUX 2005. LNCS (LNAI), vol. 3702, pp. 78–92. Springer, Heidelberg (2005). doi:10.1007/11554554_8

5. Brotherston, J.: Sequent calculus proof systems for inductive definitions. Ph.D. thesis, University of Edinburgh, November 2006
6. Brotherston, J., Bornat, R., Calcagno, C.: Cyclic proofs of program termination in separation logic. In: Proceedings of POPL-35, pp. 101–112. ACM (2008)
7. Brotherston, J., Distefano, D., Petersen, R.L.: Automated cyclic entailment proofs in separation logic. In: Bjørner, N., Sofronie-Stokkermans, V. (eds.) CADE 2011. LNCS (LNAI), vol. 6803, pp. 131–146. Springer, Heidelberg (2011). doi:10.1007/978-3-642-22438-6_12
8. Brotherston, J., Gorogiannis, N.: Cyclic abduction of inductively defined safety and termination preconditions. In: Müller-Olm, M., Seidl, H. (eds.) SAS 2014. LNCS, vol. 8723, pp. 68–84. Springer, Cham (2014). doi:10.1007/978-3-319-10936-7_5
9. Brotherston, J., Gorogiannis, N., Petersen, R.L.: A generic cyclic theorem prover. In: Jhala, R., Igarashi, A. (eds.) APLAS 2012. LNCS, vol. 7705, pp. 350–367. Springer, Heidelberg (2012). doi:10.1007/978-3-642-35182-2_25
10. Brotherston, J., Simpson, A.: Complete sequent calculi for induction and infinite descent. In: Proceedings of LICS-22, pp. 51–60. IEEE Computer Society, July 2007
11. Brotherston, J., Simpson, A.: Sequent calculi for induction and infinite descent. J. Logic Comput. **21**(6), 1177–1216 (2011)
12. Dax, C., Hofmann, M., Lange, M.: A proof system for the linear time μ-calculus. In: Arun-Kumar, S., Garg, N. (eds.) FSTTCS 2006. LNCS, vol. 4337, pp. 273–284. Springer, Heidelberg (2006). doi:10.1007/11944836_26
13. Fortier, J., Santocanale, L.: Cuts for circular proofs: semantics and cut-elimination. In: Computer Science Logic 2013 (CSL 2013), CSL 2013, 2–5 September 2013, Torino, Italy, pp. 248–262 (2013)
14. Kaye, R.: Models of Peano Arithmetic. Number 15 in Oxford Logic Guides. Oxford University Press, Oxford (1991)
15. Kolodziejczyk, L.A., Michalewski, H., Pradic, P., Skrzypczak, M.: The logical strength of Büchi's decidability theorem. In: Talbot, J.-M., Regnier, L. (eds.) 25th EACSL Annual Conference on Computer Science Logic (CSL 2016). Leibniz International Proceedings in Informatics (LIPIcs), Dagstuhl, Germany, vol. 62, pp. 36:1–36:16. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2016)
16. Martin-Löf, P.: Haupstatz for the intuitionistic theory of iterated inductive def initions. In: Fenstad, J.E. (ed.) Proceedings of the Second Scandinavian Logic Symposium, pp. 179–216. North Holland (1971)
17. Mio, M., Simpson, A.: A proof system for compositional verification of probabilistic concurrent processes. In: Pfenning, F. (ed.) FoSSaCS 2013. LNCS, vol. 7794, pp. 161–176. Springer, Heidelberg (2013). doi:10.1007/978-3-642-37075-5_11
18. Niwiński, D., Walukiewicz, I.: Games for the μ-calculus. Theoret. Comput. Sci. **163**, 99–116 (1997)
19. Santocanale, L.: A calculus of circular proofs and its categorical semantics. In: Nielsen, M., Engberg, U. (eds.) FoSSaCS 2002. LNCS, vol. 2303, pp. 357–371. Springer, Heidelberg (2002). doi:10.1007/3-540-45931-6_25
20. Simpson, S.G.: Subsystems of Second Order Arithmetic. Perspectives in Logic. Association for Symbolic Logic, New York (2009)
21. Sprenger, C., Dam, M.: A note on global induction mechanisms in a μ-calculus with explicit approximations. Theor. Inform. Appl. **37**, 365–399 (2003)
22. Sprenger, C., Dam, M.: On the structure of inductive reasoning: circular and tree-shaped proofs in the μcalculus. In: Gordon, A.D. (ed.) FoSSaCS 2003. LNCS, vol. 2620, pp. 425–440. Springer, Heidelberg (2003). doi:10.1007/3-540-36576-1_27

23. Stratulat, S.: Structural vs. cyclic induction: a report on some experiments with Coq. In: SYNASC 2016: Proceedings of the 18th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, pp. 27–34. IEEE Computer Society (2016)
24. Studer, T.: On the proof theory of the modal mu-calculus. Stud. Logica. **89**(3), 343–363 (2008)
25. Thomas, W.: Automata on infinite objects. In: van Leeuwen, J. (ed.) Handbook of Theoretical Computer Science. Formal Models and Semantics, vol. B, pp. 133–192. Elsevier Science Publishers, Amsterdam (1990)