1 Introduction

In cyclic proof systems, derivations are finite graphs annotated with sequents, rather than the common annotated finite trees. Cyclic proofs may thus present circular arguments. This calls to mind the logical fallacy of *circular reasoning*. Indeed, unfolding a cyclic proof yields an ill-founded proof tree, corresponding to infinite arguments. Soundness along infinite branches of such ill-founded proofs, and by extension cyclic proofs, cannot be reduced to truth of the system's axioms and truth-preservation of its deduction rules. The common solutions to this conundrum are to introduce well-foundedness at the level of the system's logic, rather than the level of derivation structure, or to allow infinite branches to represent coinductive arguments. As long as the proof system reasons about (co-)inductive notions, ill-founded derivations may thus still represent sound arguments. There are various features of logical systems, both logics and theories, on which cyclic proof systems may be based:

- *a domain of (co-)inductive objects:* Such systems may, for example, be arithmetical systems (Berardi and Tatsuta 2017b), type systems featuring a natural number type (**dasCircularVersionGodel2021**) or theories considering inductively defined predicates (Brotherston 2006).
- *fixed-point constructions:* The primary type of such systems are μ-calculi with explicit fixed-point quantifiers for least and greatest fixed points, such as the modal μ-calculus (Niwiński and Walukiewicz 1996), first-order μ-calculus (Sprenger and Dam 2003), higher-order fixed point logic (Kori, Tsukada and Kobayashi 2021), linear time μ-calculus (Dax, Hofmann and Lange 2006) or linear logic with fixed points (Baelde, Doumane and Saurin 2016). The suitability often extends to subsystems of the aforementioned μ-calculi, such as the alternation-free modal μ-calculus (Marti and Venema 2021), propositional dynamic (Docherty and Rowe 2019) or computational tree logic (Afshari, Leigh and Menéndez Turata 2023). A further case are systems reasoning about the inclusion relations on regular (Das and Pous 2017) and ω-regular (Hazard and Kuperberg 2022) languages, the expressions e^* and e^ω expressing a least and greatest fixed point, respectively.
- *well-founded relations:* The primary examples of such systems take a well-founded relation as a primitive, such as the <-ordering on natural num-

bers (Simpson 2017) or on ordinals (Sprenger and Dam 2003). This category also includes modal logics, such as Gödel-Löb logic (Shamkanov 2014) and Grzegorczyk logic (Savateev and Shamkanov 2021), defined over well-founded frames.

Even in cyclic proof systems for suitable logics or theories, as those described above, well-formedness according to the derivation rules is not sufficient to guarantee soundness. Instead, soundness is ensured by imposing further conditions on cyclic derivations. Such soundness conditions are discussed in great detail in Chapter 3.

Proof systems for the logics and theories characterised above whose proofs are finite trees always feature induction rules or axioms. We shall refer to such proof systems as *inductive*. The derivation rules of cyclic proof systems are often identical to those of an inductive proof system for the same logic, except for the induction rules, which the cyclic proof system either completely omits or at least greatly simplifies. There are two applications of proof systems which can benefit from such simplifications: *automatic theorem proving* and *proof theoretic investigations*.

The field of automatic theorem proving aims to develop algorithms which can prove theorems without human interaction. A major difficulty in devising such algorithms is determining at which point in a proof an induction rule should be applied. Even for human provers, finding suitable induction invariants to prove a claim can be difficult. Encoding the required 'human creativity' into an algorithm naturally poses a difficult problem. This problem can be circumvented completely by developing automated theorem provers in terms of a cyclic proof system without induction rules. Indeed, multiple experimental theorem provers following this approach have already been developed (see Brotherston, Distefano and Petersen 2011; Brotherston, Gorogiannis and Petersen 2012; Tellez and Brotherston 2017).

Another application of cyclic proof systems is carrying out proof theoretic investigations of logics and theories with features as described above. Induction axioms are often non-logical rules, not directly tied to logical connectives, and can thus 'disturb' the 'homogeneity' of a proof system. This, for example, is exhibited by the fact that Cuts on induction axioms can not be eliminated in some theories. Cyclic proof systems, which forgo induction axioms in favour of cycles, can be better suited to certain kinds of proof theoretic applications. So far, cyclic proof systems have proven especially useful to deriving interpolation results (see Shamkanov 2014; Savateev and Shamkanov 2021; Marti and Venema 2021; Afshari, Leigh and Menéndez Turata 2021; Afshari and Leigh 2022). They have also been used to derive results about non-cyclic proof systems, such as the Cut-free completeness of an axiomatisation of the modal μ-calculus (Afshari and Leigh 2016).

Lastly, cyclic proof systems are of interest as objects of proof theoretic study in their own right. This is the perspective assumed in this thesis. Cyclic proof theory is a rather young field, arguably originating in the works of Niwiński and Walukiewicz (1996) and Sprenger and Dam (2003). Some of the most basic questions, such as general strategies for Cut-elimination in cyclic proofs, remain open.

2 Cyclic Arithmetics

2.1 Simple Cyclic Heyting Arithmetic

We begin by introducing a cyclic proof system for Heyting arithmetic, using said system to illustrate some wider aspects of cyclic proof theory. We call this system *simple* cyclic Heyting arithmetic because some of its combinatorial aspects are simpler than those of the cyclic proof systems for Heyting arithmetic we introduce in Section 2.3.

Heyting arithmetic employs the language of first-order arithmetic, which is given below. We fix a countable set of variables $\textsc{Var}$ and write $\mathrm{FV}(\varphi)$ for the set of *free variables* which occur unbound in a formula φ.

$$
\begin{aligned}
s, t \in \textsc{Term} &::= x \mid 0 \mid Ss \mid s + t \mid s \cdot t && x \in \textsc{Var} \\
\varphi, \psi \in \textsc{Form} &::= s = t \mid \bot \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \to \psi \mid \forall x.\varphi \mid \exists x.\varphi
\end{aligned}
$$

Denote by $[t/x]$ the usual *substitution operation*, replacing all free occurrences of the variable x by the term t. On formulas, this is a partial operation, $\varphi[t/x]$ being undefined if the free variables in t are not distinct from the bound variables in φ. Henceforth, writing $\varphi[t/x]$ will double as an assertion of the resulting formula being defined.

We begin by recalling an inductive proof system for Heyting arithmetic, i.e. one presented in the common manner of finite derivation trees.

Definition (Heyting arithmetic) The *sequents* of *Heyting arithmetic* are expressions $\Gamma \Rightarrow \delta$ where Γ is a finite set of formulas and δ a single formula of first-order arithmetic. Write Γ, φ for $\Gamma \cup \{\varphi\}$ and Γ, Γ' for $\Gamma \cup \Gamma'$. The *derivation rules* of HA comprise of the following choice of standard rules for classical first-order logic with equality,

$$\text{Ax}\,\frac{}{\Gamma, \delta \Rightarrow \delta} \qquad \to\text{L}\,\frac{\Gamma, \varphi \Rightarrow \delta \quad \Gamma \Rightarrow \psi}{\Gamma, \varphi \to \psi \Rightarrow \delta} \qquad \to\text{R}\,\frac{\Gamma, \varphi \Rightarrow \psi}{\Gamma \Rightarrow \varphi \to \psi}$$

$$\wedge\text{L}\,\frac{\Gamma, \varphi, \psi \Rightarrow \delta}{\Gamma, \varphi \wedge \psi \Rightarrow \delta} \qquad \wedge\text{R}\,\frac{\Gamma \Rightarrow \varphi \quad \Gamma \Rightarrow \psi}{\Gamma \Rightarrow \varphi \wedge \psi}$$

$$\vee\text{L}\,\frac{\Gamma, \varphi \Rightarrow \delta \quad \Gamma, \psi \Rightarrow \delta}{\Gamma, \varphi \vee \psi \Rightarrow \delta} \qquad \vee\text{R}\,\frac{\Gamma \Rightarrow \varphi_i}{\Gamma \Rightarrow \varphi_1 \vee \varphi_2} \qquad \forall\text{L}\,\frac{\Gamma, \varphi[t/x] \Rightarrow \delta}{\Gamma, \forall x.\varphi \Rightarrow \delta}$$

$$\forall\text{R}\,\frac{\Gamma \Rightarrow \varphi \quad x \notin \text{FV}(\Gamma, \forall x.\varphi)}{\Gamma \Rightarrow \forall x.\varphi} \qquad \exists\text{L}\,\frac{\Gamma, \varphi \Rightarrow \delta \quad x \notin \text{FV}(\Gamma, \exists x.\varphi, \delta)}{\Gamma, \exists x.\varphi \Rightarrow \delta}$$

$$\exists\text{R}\,\frac{\Gamma \Rightarrow \varphi[t/x]}{\Gamma \Rightarrow \exists x.\varphi} \qquad \bot\text{L}\,\frac{}{\Gamma, \bot \Rightarrow \delta}$$

$$=\text{L}\,\frac{\Gamma[x/y] \Rightarrow \delta[x/y] \quad x, y \notin \text{FV}(s, t)}{\Gamma[s/x, t/y], s = t \Rightarrow \delta[s/x, t/y]} \qquad =\text{R}\,\frac{}{\Gamma \Rightarrow t = t}$$

with the following two structural rules,

$$\text{Wk}\,\frac{\Gamma \Rightarrow \delta}{\Gamma, \Gamma' \Rightarrow \delta} \qquad \text{Cut}\,\frac{\Gamma \Rightarrow \varphi \quad \Gamma, \varphi \Rightarrow \delta}{\Gamma \Rightarrow \delta}$$

the following axiomatic sequents which characterise the function symbols of first-order arithmetic,

$$\begin{array}{lll} \Rightarrow 0 \neq St & Ss = St \Rightarrow s = t & \Rightarrow s + 0 = 0 \\ \Rightarrow s + St = S(s + t) & \qquad\Rightarrow s \cdot 0 = 0 & \Rightarrow s \cdot St = (s \cdot t) + s \end{array}$$

and the axiomatic sequents of *induction*, for any formula φ,

$$\varphi(0), \forall x.\varphi(x) \to \varphi(Sx) \Rightarrow \varphi(s).$$

If a proof with endsequent $\Gamma \Rightarrow \delta$ exists then $\Gamma \Rightarrow \delta$ is *provable in Heyting arithmetic*, denoted by $\text{HA} \vdash \Gamma \Rightarrow \delta$. ⌟

To illustrate the contrast with cyclic proof systems, it is helpful to consider the representation of proofs in Heyting arithmetic. Formally, a proof in HA is a pair $\Pi = (T, \lambda)$ consisting of a finite tree T and a function $\lambda : \text{Node}(T) \to$ Seq labelling the nodes in accordance with the derivation rules given above.

Considering the conclusions of rules without premises as axiomatic, if $t \in \mathrm{Leaf}(T)$ then $\lambda(t)$ is an axiomatic sequent.

The underlying structure of cyclic proofs is somewhat more complicated. Often, this structure is described simply as 'finite graphs'. However, in practice, a stricter notion consisting of a finite tree with 'added cycles' often proves more useful, retaining common notions, such as 'root', 'leaf' and 'subtree'. A *cyclic tree* is a pair $C = (T, \beta)$ consisting of a finite tree T and a partial function $\beta\colon \mathrm{Leaf}(T) \to \mathrm{Inner}(T)$ mapping some leaves of T onto inner nodes of T. A leaf $b \in \mathrm{dom}(\beta)$ is called a *bud* and $\beta(b)$ its *companion*.

Commonly, a further restriction is imposed on the cyclic trees underlying cyclic proofs. A cyclic tree is said to be in *cycle normal form* if for every bud $b \in \mathrm{dom}(\beta)$ its companion $\beta(b)$ occurs along the path from the tree's root to b. This restriction has various useful consequences. For example, all buds of an inner node must be located in the subtree above it, easing recursive proof translations. Furthermore, trees in cycle normal form possess a well-defined notion of the 'inside' of a cycle: the path from $\beta(t)$ to t for $t \in \mathrm{dom}(\beta)$. Every cyclic tree can be unfolded into a cyclic tree in cycle normal form, although this may cause a super-exponential increase in the number of nodes (see Brotherston 2006, Theorem 6.3.6).

We proceed by defining a cyclic proof system CHA for Heyting arithmetic. It eschews the induction axioms in favour of *cycles* which allow leaves to be considered closed if they are labelled by a sequent that has previously appeared on the branch between the root and said node.

Definition (CHA pre-proofs) The sequents of *cyclic Heyting arithmetic* are the same as in regular Heyting arithmetic. A *pre-proof* of CHA is a pair $\Pi = (C, \lambda)$ consisting of a cyclic tree $C = (T, \beta)$ and a labelling function $\lambda : \mathrm{Node}(T) \to \textsc{Seq}$. The labelling function λ labels the nodes of T with sequents such that

- inner nodes are labelled according to the derivation rules of HA or the CHA-specific derivation rules

$$\textsc{Case}_x\frac{\Gamma[0/x] \Rightarrow \delta[0/x] \quad \Gamma[Sx/x] \Rightarrow \delta[Sx/x]}{\Gamma \Rightarrow \delta}$$

$$\textsc{Subst}\frac{\Gamma \Rightarrow \delta}{\Gamma[t/x] \Rightarrow \delta[t/x]}$$

  in which $x \notin \mathrm{FV}(\Gamma, \delta)$ for $\textsc{Case}_x$ and $x \notin \mathrm{FV}(t)$ for $\textsc{Subst}$.
- leaves, with the exception of buds, must be labelled with HA axioms, excluding the induction axioms

- each bud and companion share labels, i.e. $\lambda(s) = \lambda(\beta(s))$ for $s \in \mathrm{dom}(\beta)$.

For an example of a CHA pre-proof, consider Figure 1. It concludes the induction scheme of HA. We employ the shorthand $\overline{\varphi} := \varphi(0), \forall x.\varphi(x) \to \varphi(Sx)$ for the left-hand side of the induction scheme. The two sequents marked with a $\star$ form a cycle.

$$
\begin{prooftree}
\AxiomC{}
\LeftLabel{\scriptsize AX}
\UnaryInfC{$\overline{\varphi}, \varphi(0) \Rightarrow \varphi(0)$}
\AxiomC{$\overline{\varphi} \Rightarrow \varphi(x) \ \star$}
\LeftLabel{\scriptsize WK}
\UnaryInfC{$\overline{\varphi} \Rightarrow \varphi(x), \varphi(Sx)$}
\AxiomC{}
\RightLabel{\scriptsize AX}
\UnaryInfC{$\overline{\varphi}, \varphi(Sx) \Rightarrow \varphi(Sx)$}
\RightLabel{\scriptsize $\forall$L,$\to$L}
\BinaryInfC{$\overline{\varphi}, \forall x.\varphi(x) \to \varphi(Sx) \Rightarrow \varphi(Sx)$}
\LeftLabel{\scriptsize CASE$_x$}
\BinaryInfC{$\overline{\varphi} \Rightarrow \varphi(x) \ \star$}
\LeftLabel{\scriptsize SUBST}
\UnaryInfC{$\overline{\varphi} \Rightarrow \varphi(s)$}
\end{prooftree}
$$

Figure 1: A CHA pre-proof of the induction axiom, $\star$ marking a cycle

However, not every CHA pre-proof is a sound argument. For a counter-example, consider Figure 2, in which an 'argument' concluding the inconsistency $\Rightarrow \bot$ is made.

$$
\begin{prooftree}
\AxiomC{}
\LeftLabel{\scriptsize $\bot$L}
\UnaryInfC{$\bot \Rightarrow \bot$}
\AxiomC{$\Rightarrow \bot \ \star$}
\LeftLabel{\scriptsize CUT}
\BinaryInfC{$\Rightarrow \bot \ \star$}
\end{prooftree}
$$

Figure 2: An unsound CHA pre-proof, $\star$ marking a cycle

To ensure soundness of CHA proofs, an additional condition beyond well-formedness, called a *soundness condition*, must be imposed. The soundness condition given in **??** is called a *global trace condition*. We consider alternative soundness conditions for cyclic Heyting arithmetic in Chapter 3.

DEFINITION (CHA PROOFS) Let $\Pi = (C, \lambda)$ be a pre-proof. An *infinite branch* through Π is an infinite sequence $t \in \mathrm{Node}(T)^\omega$ with t_0 being the root node of T and such that for each $i \in \omega$ either $t_{i+1} \in \mathrm{Chld}(t_i)$ or $t_i \in \mathrm{dom}(\beta)$ and $t_{i+1} = \beta(t_i)$. This induces an infinite sequence $(\Gamma_i \Rightarrow \delta_i)_{i\in\omega}$ of sequents with $\lambda(t_i) = \Gamma_i \Rightarrow \Delta_i$, which we use interchangeably with $(t_i)_{i\in\omega}$ to denote an infinite branch. A variable x is said to have *a trace along* $(\Gamma_i \Rightarrow \delta_i)_{i\in\omega}$ if there exists an $n \in \omega$ such that $x \in \mathrm{FV}(\Gamma_i, \delta_i)$ for all $i \geq n$. Such a trace on x is said to be *progressing* if it passes through instances of the CASE$_x$-rule (for the same variable x) infinitely often.

A pre-proof Π is a *proof* in cyclic Heyting arithmetic if for every infinite branch through Π there exists a variable which has a progressing trace along it. $\mathrm{CHA} \vdash \Gamma \Rightarrow \delta$ denotes provability in CHA. ⌟

Observe that the pre-proof of $\Rightarrow \bot$ in Figure 2 is no CHA proof because it does not contain any free variables. The pre-proof of the induction scheme in Figure 1 is a proper proof: It only has one infinite branch, obtained by following the $\star$-cycle indefinitely. This branch has an x-trace, starting above the Subst-application, which passes through the Case_x-rule infinitely often.

From the perspective of proof-as-programs, cyclic proofs correspond to recursively defined functions, the soundness conditions corresponding to termination conditions. This connection has been used to bring notions and results from the field of program termination (Lee, Jones and Ben-Amram 2001) to the setting of cyclic proof theory (Nollet, Saurin and Tasson 2019; Afshari and Wehr 2022).

The global trace condition is the most common soundness condition in cyclic proof theory. It usually follows a common template: Along every infinite branch, a syntactic feature (such as a term, fixed-point quantifier, formula, ...) can be traced which *progresses* (e.g. decreases, is unfolded, is principal to certain derivation rules, ...) infinitely often. Based on this observation, abstract notions of trace, progress and cyclic proof have been formulated which allow the study of cyclic proofs from a general perspective (Brotherston 2006; Afshari and Wehr 2022). We discuss the latter framework in ??. It is relied on in Paper I to carry out a construction for many cyclic proof systems at once.

The system CHA is merely one possible cyclic proof system for Heyting arithmetic. Even when restricting one's attention to cyclic proof systems with global trace conditions, there are two other renditions of Heyting arithmetic as a cyclic proof system in the literature, both of which we present in Section 2.3.

The cyclic proof system CHA can be transformed into a cyclic proof system CPA of Heyting arithmetic via the usual broadening of the right-hand side of sequents to a finite set Δ. The global trace condition, including the employed notions of branch, trace and progress, remain unchanged.

The purpose of the global trace condition in ?? is most easily illustrated by proving the soundness of CHA proofs over the standard model of the natural numbers, i.e. the structure ω of first-order arithmetic whose domain are the natural numbers and which interprets the function symbols in the usual manner.

Theorem (Soundness) If $\mathrm{CHA} \vdash \Gamma \Rightarrow \delta$ then $\omega \vDash \Gamma \Rightarrow \delta$.

Proof Let Π be a CHA proof of $\Gamma \Rightarrow \delta$ and suppose, towards contradiction, that there is an assignment $\rho : \textsc{Var} \to \omega$ such that $\rho \not\models \Gamma \Rightarrow \delta$. Consider the last CHA-rule applied to derive $\Gamma \Rightarrow \delta$ in Π. There must exist a premise $\Gamma_1 \Rightarrow \delta_1$ and an assignment ρ_1 such that $\rho_1 \not\models \Gamma_1 \Rightarrow \delta_1$. We treat a few illustrative cases:

$\wedge$R: Then $\delta = \varphi_0 \wedge \varphi_1$ and the premises of the rule are $\Gamma \Rightarrow \varphi$ and $\Gamma \Rightarrow \psi$. Fix $\rho_1 := \rho$. If $\rho \not\models \Gamma \Rightarrow \delta$ then $\rho \models \Gamma$ and $\rho \not\models \delta$. Thus there must be $i \in \{0, 1\}$ such that $\rho \not\models \varphi_i$ and thus $\rho \not\models \varphi_i$.

Subst: Then $\Gamma = \Gamma'[t/x]$ and $\delta = \delta'[t/x]$ for $x \notin \mathrm{FV}(t)$. Furthermore, the premise of the rule is $\Gamma' \Rightarrow \delta'$. Fix $\rho_1 := \rho[x \mapsto t^\rho]$, i.e. the environment which assigning the same values to $y \neq x$ as ρ and the evaluation of t under ρ to x. Observe that for every term s, $(s[t/x])^\rho = s^{\rho_1}$. Thus, $\rho \not\models \Gamma[t/x] \Rightarrow \delta[t/x]$ entails $\rho_1 \not\models \Gamma \Rightarrow \delta$.

$\forall$R: Then $\delta = \forall x.\varphi$ with $x \notin \mathrm{FV}(\Gamma, \delta)$ and the premise is $\Gamma \Rightarrow \varphi$. By $\rho \not\models \Gamma \Rightarrow \forall x.\varphi$ there must exist $n \in \omega$ such that $\rho[x \mapsto n] \not\models \varphi$. Hence, fixing $\rho_1 := \rho[x \mapsto n]$, obtains $\rho_1 \not\models \Gamma \Rightarrow \varphi$ because $x \notin \mathrm{FV}(\Gamma, \delta)$.

$\forall$L: Then $\Gamma = \Gamma', \forall x.\varphi$ and the premise is $\Gamma, \varphi[t/x] \Rightarrow \delta$. From $\rho \not\models \Gamma, \forall x.\varphi$ it follows that $\rho \models \forall x.\varphi$ and thus $\rho[x \mapsto t^\rho] \models \varphi$, or equivalently, $\rho \models \varphi[t/x]$. Hence, fixing $\rho_1 := \rho$, it follows that $\rho_1 \not\models \Gamma', \varphi[t/x] \Rightarrow \delta$.

=L: Then $\Gamma = \Gamma'[s/x, t/y], s = t$ and $\delta = \delta'[s/x, t/y]$ with $x, y \notin \mathrm{FV}(s, t)$ and the premise is $\Gamma'[t/x, s/y] \Rightarrow \delta'[t/x, s/y]$. From $\rho \not\models \Gamma \Rightarrow \delta$ it follows that $\rho \models s = t$ and hence $s^\rho = t^\rho$. Hence, for any term u and $v \in \{x, y\}$, $(u[s/v])^\rho = (u[t/x])^\rho$ and thus $\rho \not\models \Gamma'[s/x, t/y], s = t \Rightarrow \delta'[s/x, t/y]$ entails $\rho \not\models \Gamma'[t/x, s/y] \Rightarrow \delta'[t/x, s/y]$.

$\textsc{Case}_x$: Then $x \in \mathrm{FV}(\Gamma, \delta)$. Observe that either $\rho(x) = 0$ or $\rho(x) = n + 1$. In the former case, $\rho \not\models \Gamma[0/x] \Rightarrow \delta[0/x]$, in the latter case $\rho[x \mapsto n] \not\models \Gamma[Sx/x] \Rightarrow \delta[Sx/x]$.

The remaining cases are analogous to those considered above. This argument can be applied to any non-axiomatic node of Π. The contradicted premises arrived at cannot be axioms as they cannot be contradicted in the standard model. Iterating this argument, 'following' buds to their companions, yields an infinite branch $(\Gamma_i \Rightarrow \delta_i)_{i\in\omega}$ through Π and a sequence $(\rho_i)_{i\in\omega}$ of contradicting assignments such that $\rho_i \not\models \Gamma_i \Rightarrow \delta_i$. As Π is a proof, there exists a progressing x-trace along $(\Gamma_i \Rightarrow \delta_i)_{i\in\omega}$ (w.l.o.g. we assume it starts at $\Gamma_0 \Rightarrow \delta_0$). Now consider the sequence $(\rho_i(x))_{i\in\omega}$ in ω. From the cases above it follows that an increase, i.e. $\rho_i(x) < \rho_{i+1}(x)$, can only occur in the cases of $\forall$R and $\exists$L with x being the bound variable. However, as x is free in all Γ_i, δ_i, this cannot occur along the branch. On the other hand, decreases, i.e. $\rho_i(x) > \rho_{i+1}(x)$, take place whenever the right-hand side of a $\textsc{Case}_x$-rule is passed. As this takes place infinitely often, $(\rho_i(x))_{i\in\omega}$ is a never increasing, infinitely decreasing sequence of natural numbers, which cannot exist. Hence, the conclusion of Π could not have been invalid to begin with. $\square$

We have stated that the common strategies to incorporate the infinite branches of ill-founded proofs into sound arguments was to introduce well-foundedness or coinduction at the level of the logic. CHA takes the former approach, ensuring that all infinite branches of a CHA proof produce an infinitely decreasing sequence and can thus be disregarded when considering soundness. Other systems, such as μ-calculi, allow for coinductive arguments, ensuring that greatest fixed points are unfolded infinitely along infinite branches

The proof strategy in ?? is employed throughout the literature to prove soundness of cyclic proof systems with a global trace condition. Often, the demonstrated decreasing sequence is a sequence of ordinals, rather than just natural numbers. Brotherston (2006, Section 4.2.1) gives a general account of this style of soundness proof. [TODO: We give a better proof!] It should be noted the soundness proof is highly classical: deducing the existence of a contradicted premise requires variants of the law of the excluded middle and 'constructing' the contradicted branch by iterating the argument requires a variant of the axiom of choice. A corollary of the equivalences between CHA and HA proven in Paper II is a fully constructive soundness proof of CHA.

2.2 Relating Proof Systems

This section discusses how cyclic proof systems may be related to other kinds of proof systems. We continue to work with CHA as the example. The soundness result in ?? does not contribute towards proving the equivalence between the 'inductive' proof system HA and the cyclic proof system CHA. Instead, this equivalence is witnessed by proof theoretic translations back and forth. One of the two directions can be proven using the notions introduced so far.

Theorem If $\mathrm{HA} \vdash \Gamma \Rightarrow \delta$ then $\mathrm{CHA} \vdash \Gamma \Rightarrow \delta$.

Proof Recall that Figure 1 gives CHA proofs for every instance of the induction scheme of HA. Now, suppose $\mathrm{HA} \vdash \Gamma \Rightarrow \delta$. A HA proof Π differs from a CHA pre-proof only by the fact that some of its leaves may be labelled with instances of the induction schemes. Thus, a CHA pre-proof Π' of $\Gamma \Rightarrow \delta$ may be obtained by 'grafting' copies of the CHA proof in Figure 1 onto each such leaf of Π. As there are no cycles in Π, every infinite branch of Π' must follow the $\star$-cycle of one of the 'grafted on' induction scheme proofs and thus has a successful trace. Hence Π' is a CHA proof witnessing $\mathrm{CHA} \vdash \Gamma \Rightarrow \delta$. $\square$

Whereas the translation given in ?? is rather simple, the converse direction of the equivalence is much more complicated. It was simultaneously obtained by Simpson (2017) and Berardi and Tatsuta (2017a). It is also the main concern of [TODO: Paper II]. We thus contend ourselves with simply stating the result, referring the reader to Paper II for its proof.

Fact If CHA $\vdash \Gamma \Rightarrow \delta$ then HA $\vdash \Gamma \Rightarrow \delta$.

In cyclic proof systems for logics, rather than first-order theories like Heyting or Peano arithmetic, the equivalence to inductive proof systems is usually established by proving the soundness and completeness of the cyclic proof system directly in terms of a canonical semantics of the logic. When comparing the proofs of soundness and completeness between the inductive and cyclic proof systems, cyclic proof systems often require more intricate proofs of soundness (as witnessed by Theorem 2.4) but allow for simpler completeness proofs. One intuition behind this phenomenon is that ill-founded proofs (and hence cyclic proofs) are *coinductive* objects whereas inductive proofs are *inductive* objects. Framed in the language of category theory, algebras ease the construction of morphisms 'out of them' via recursion (corresponding to the 'direction' of soundness) whereas coalgebras ease the construction of morphisms 'into them' via corecursion (corresponding to the 'direction' of completeness).

For some cyclic proof systems, it is not yet known whether they are equivalent to the 'corresponding' inductive proof system. Brotherston (2006) puts forward a family of inductive (LKID) and cyclic (CLKID$^{\omega}$) proof systems of first-order logic extended with concrete inductive definitions. He conjectures but does not prove the equivalence of LKID and CLKID$^{\omega}$, a claim which is now referred to as the *Brotherston-Simpson conjecture*. Berardi and Tatsuta (2017a) give a partial solution to the conjecture: If the systems feature an inductive natural number predicate $N\,t$ and the functions of first-order arithmetic, i.e. if the systems contain Peano arithmetic, then LKID and CLKID$^{\omega}$ prove the same theorem. This result has been extended to intuitionistic variants of LKID and CLKID$^{\omega}$ containing Heyting arithmetic in (Berardi and Tatsuta 2017b). Further, **tatsutaClassicalSystemMartinLof2019** have shown that for a 'weaker' inductive predicate, the two systems are not equivalent. However, a full classification of which inductive predicates induce equivalence has not yet been given.

Cyclic proofs can be viewed as finite representations of ill-founded proofs. In the setting of first-order arithmetic, such ill-founded proofs are interesting in their own right.

Definition (CHA ∞-proofs) The *sequents* of CHA ∞-pre-proofs are the same as those of Peano arithmetic. An *∞-pre-proof* is a pair $\Pi = (T, \lambda)$ consisting of a (possibly) ill-founded tree T and a labelling function $\lambda : \mathrm{Node}(T) \to \mathrm{Seq}$. The labelling function λ labels the nodes of T with sequents such that

- inner nodes are labelled according to the derivation rules of HA or the CHA-specific derivation rule

$$\textsc{Case}_x\,\frac{\Gamma[0/x] \Rightarrow \delta[0/x] \quad \Gamma[Sx/x] \Rightarrow \delta[Sx/x]}{\Gamma \Rightarrow \delta} \qquad \textsc{Subst}\,\frac{\Gamma \Rightarrow \delta}{\Gamma[t/x] \Rightarrow \delta[t/x]}$$

  in which $x \notin \mathrm{FV}(\Gamma, \delta)$ for $\textsc{Case}_x$ and $x \notin \mathrm{FV}(t)$ for $\textsc{Subst}$.
- leaves labelled with HA axioms, excluding the induction axioms.

A CHA ∞-pre-proof Π is a CHA *∞-proof* if for every infinite branch through Π there exists a variable which has a progressing trace along it. Provability via CHA ∞-proofs is denoted $\mathrm{CHA} \vdash^{\infty} \Gamma \Rightarrow \delta$. ⌟

Every CHA pre-proof can be 'unfolded' into an CHA ∞-pre-proof by continuously replacing each bud with the subtree above, and including, its companion. Because the soundness condition imposed on CHA proofs and CHA ∞-proofs is 'the same', this method transforms CHA proofs into ∞-proofs. The CHA proofs corresponds to a class of ∞-proofs: the *regular* CHA ∞-proofs, i.e. those with finitely many distinct subproofs. The notion of ∞-proof is rather general and can be considered in the context of any cyclic proof system. Conversely, the regular proofs of an ∞-proof system always induce a cyclic proof systems.

From Gödel's incompleteness theorems, it follows that HA (and thus CHA) is incomplete, i.e. that there are statements satisfied by the standard model ω of arithmetic which are not provable in HA. Completeness can be achieved by extending HA with an infinitary ω-rule which allows universal statements to be deduced from derivations of the statement for every numeral $S^n 0$ for $n \in \omega$.

$$\omega\,\frac{\Gamma \Rightarrow \varphi[0/x] \quad \Gamma \Rightarrow \varphi[S0/x] \quad \ldots \quad \Gamma \Rightarrow \varphi[S^n 0/x] \quad \ldots}{\Gamma \Rightarrow \forall x.\varphi}$$

Simpson (2017, Theorem 4) observes that the ω-rule is admissible in ∞-proofs.

Proposition The ω-rule is admissible in ∞-proofs.

Proof Suppose there were ∞-proofs Π_n of $\Gamma \Rightarrow \varphi[S^n 0/x]$ for every $n \in \omega$. Consider the ∞-pre-proof Π below, in which w.l.o.g. $x \notin \mathrm{FV}(\Gamma, \forall x.\varphi)$.

$$
\dfrac{\dfrac{\dfrac{\Pi_0}{\Gamma \Rightarrow \varphi[0/x]} \qquad \dfrac{\dfrac{\Pi_1}{\Gamma \Rightarrow \varphi[S0/x]} \qquad \begin{array}{c} \dfrac{\dfrac{\Pi_n}{\Gamma \Rightarrow \varphi[S^n 0/x]} \qquad \dfrac{\vdots}{\Gamma \Rightarrow \varphi[S^{n+1}x/x]}}{\Gamma \Rightarrow \varphi[S^n x/x]}\,\textsc{Case}_x \\ \vdots \\ \Gamma \Rightarrow \varphi[S^2 x/x] \end{array}}{\Gamma \Rightarrow \varphi[Sx/x]}\,\textsc{Case}_x}{\Gamma \Rightarrow \varphi}\,\textsc{Case}_x}{\Gamma \Rightarrow \forall x.\varphi}\,\forall\mathrm{R}
$$

It remains to argue that Π is a proof. Thus, consider an infinite path through it. Such a path must either infinitely pass through the right-hand sides of the $\textsc{Case}_x$-applications or enter one of the Π_n. Per assumption, every branch through the Π_n has a progressing trace. As the notion of progressing trace is closed under taking prefixes, all the branches through Π entering one of the Π_n exhibit one. The branch passing infinitely many right-hand sides of $\textsc{Case}_x$ clearly has a progressing x-trace. Hence Π is a proof. □

Corollary (Completeness) If $\omega \vDash \Gamma \Rightarrow \delta$ then $\mathrm{CHA} \vdash^{\infty} \Gamma \Rightarrow \delta$.

For the cyclic proofs for first-order logic with inductive definitions considered by Brotherston (2006), a similar mismatch is exhibited, the ∞-proofs being complete for standard second-order models. However, this disagreement between cyclic and ∞-proofs is not universal. For example, in modal logics like the modal μ-calculus (Niwiński and Walukiewicz 1996), Gödel-Löb logic (Shamkanov 2014) and Grzegorczyk logic (Savateev and Shamkanov 2021), cyclic proofs and ∞-proofs prove exactly the same theorems. More generally, cyclic and ∞-proof systems in which only finitely many distinct sequents can occur in a proof are always equivalent (see Wehr 2021, Theorem 4.6). This result applies to many proof systems which exhibit a subformula property, such as the modal logics listed above.

Returning to the setting of first-order arithmetic, Heyting and true arithmetic corresponding to regular and unrestricted ∞-proofs, respectively, naturally raises the question whether correspondences between other classes of ∞-proofs and fragments of first-order arithmetic can be found. An interesting extension of regular trees are the classes of hyper-regular trees generated by higher-order recursion schemes as considered in Ong (2006). The article proves that acceptance of ω-regular conditions, such as the global trace condition, on such hyper-regular trees is decidable, extending the property of decidable proof checking to hyper-regular ∞-proofs. However, the trees considered may only employ a finite supply of labels, corresponding to the case of proof systems with finitely many sequents discussed in the previous paragraph. Hence, higher-order recursion system generated ∞-proofs (for virtually every cyclic proof system in the literature) prove the same theorems as cyclic proofs. This is the only other such result we are aware of.

2.3 More Cyclic Arithmetics

Simpson (2017) presents a proof system over the language of first-order arithmetic extended by a relation $s < t$. Traces in his system are sequences of terms appearing along the branch which are, at each 'step', either equal or ordered by a formula $s < t$ to the left of $\Rightarrow$. Each $<$-step is considered a progression. Berardi and Tatsuta (2017a) consider a system of first-order arithmetic extended by inductively defined predicates, including a one-place predicate $N\,t$ asserting that t denotes a natural number. Traces are sequences of inductively defined predicates appearing to the left of $\Rightarrow$. Progress takes place whenever a case-distinction rule is applied to the traced predicate, the effect of a case-distinction on $N\,t$ being analogous to that of the $\textsc{Case}_x$-rule of Definition 2.2. An important difference between our notion of trace and those employed in the two aforementioned articles is that it does not 'respect' the $\textsc{Subst}$ rule, i.e. if an x-trace passing through a $\textsc{Subst}$-instance which replaces x by a fresh y does not 'switch focus' to y but is simply terminated.

2.3.1 Simpson-style Cyclic Heyting Arithmetic

The term and formula languages of $\mathrm{CHA}_<$ are given below. The formula language is non-standard, treating inequality $s < t$ as a primitive, rather than a defined notion. As will become clear below, this eases the definition of the global trace condition of $\mathrm{CHA}_<$.

$$
\begin{aligned}
s, t \in \textsc{Term} &::= x \mid 0 \mid Ss \mid s + t \mid s \cdot t \\
\varphi, \psi \in \textsc{Form} &::= s = t \mid s < t \mid \bot \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \to \psi \mid \forall x.\varphi \mid \exists x.\varphi
\end{aligned}
$$

We still denote by $[t/x]$ the *substitution operation* described in Section 2.1.

Definition The *sequents* of $\mathrm{CHA}_<$ are expressions $\Gamma \Rightarrow \delta$ where Γ is a finite sets of formulas and δ a single formula. Write Γ, φ for $\Gamma \cup \{\varphi\}$ and Γ, Γ' for $\Gamma \cup \Gamma'$. The *derivation rules* of $\mathrm{CHA}_<$ comprise of the following choice of standard rules for intuitionistic first-order logic,

$$\text{Ax}\,\frac{}{\Gamma, \delta \Rightarrow \delta} \qquad \to\text{L}\,\frac{\Gamma, \varphi \Rightarrow \delta \quad \Gamma \Rightarrow \psi}{\Gamma, \varphi \to \psi \Rightarrow \delta} \qquad \to\text{R}\,\frac{\Gamma, \varphi \Rightarrow \psi}{\Gamma \Rightarrow \varphi \to \psi}$$

$$\wedge\text{L}\,\frac{\Gamma, \varphi, \psi \Rightarrow \delta}{\Gamma, \varphi \wedge \psi \Rightarrow \delta} \qquad \wedge\text{R}\,\frac{\Gamma \Rightarrow \varphi \quad \Gamma \Rightarrow \psi}{\Gamma \Rightarrow \varphi \wedge \psi}$$

$$\vee\text{L}\,\frac{\Gamma, \varphi \Rightarrow \delta \quad \Gamma, \psi \Rightarrow \delta}{\Gamma, \varphi \vee \psi \Rightarrow \delta} \qquad \vee\text{R}\,\frac{\Gamma \Rightarrow \varphi_i}{\Gamma \Rightarrow \varphi_1 \vee \varphi_2} \qquad \forall\text{L}\,\frac{\Gamma, \varphi[t/x] \Rightarrow \delta}{\Gamma, \forall x.\varphi \Rightarrow \delta}$$

$$\forall\text{R}\,\frac{\Gamma \Rightarrow \varphi \quad x \notin \mathrm{FV}(\Gamma, \forall x.\varphi)}{\Gamma \Rightarrow \forall x.\varphi} \qquad \exists\text{L}\,\frac{\Gamma, \varphi \Rightarrow \delta \quad x \notin \mathrm{FV}(\Gamma, \exists x.\varphi, \delta)}{\Gamma, \exists x.\varphi \Rightarrow \delta}$$

$$\exists\text{R}\,\frac{\Gamma \Rightarrow \varphi[t/x]}{\Gamma \Rightarrow \exists x.\varphi} \qquad \bot\text{L}\,\frac{}{\Gamma, \bot \Rightarrow \delta}$$

$$=\!\text{L}\,\frac{\Gamma[t/x, s/y] \Rightarrow \delta[t/x, s/y] \quad x, y \notin \mathrm{FV}(s, t)}{\Gamma[s/x, t/y], s = t \Rightarrow \delta[s/x, t/y]} \qquad =\!\text{R}\,\frac{}{\Gamma \Rightarrow t = t}$$

with the following structural rules,

$$\text{Wk}\,\frac{\Gamma \Rightarrow \delta}{\Gamma, \Gamma' \Rightarrow \delta} \qquad \text{Cut}\,\frac{\Gamma \Rightarrow \varphi \quad \Gamma, \varphi \Rightarrow \delta}{\Gamma \Rightarrow \delta} \qquad \text{Sub}\,\frac{\Gamma \Rightarrow \delta}{\Gamma[s/x] \Rightarrow \delta[s/x]}$$

with $x \notin \mathrm{FV}(t)$ for Subst and the following arithmetic-specific axioms

$$\begin{array}{rlrl}
s < t, t < u \Rightarrow & s < u & s < t \Rightarrow & Ss < St \\
\Rightarrow & s + St = S(s + t) & s < t, t < s \Rightarrow & \\
\Rightarrow & s < t \vee s = t \vee t < s & \Rightarrow & t \cdot 0 = 0 \\
s < t, t < Ss \Rightarrow & & \Rightarrow & t < St \\
\Rightarrow & s \cdot St = (s \cdot t) + s & t < 0 \Rightarrow & \\
\Rightarrow & t + 0 = t & &
\end{array}$$

and the arithmetic-specific derivation rule

$$\text{S}\,\frac{\Gamma, t = Sx \Rightarrow \delta \quad x \text{ fresh}}{\Gamma, 0 < t \Rightarrow \delta}$$

Definition (**Trace**) A term t *occurs* in a sequent $\Gamma \Rightarrow \delta$ if it appears, possibly as a subterm of another term, in a formula in Γ or δ. Write $\textsc{Term}(\Gamma \Rightarrow \delta)$ for the set of terms occurring in $\Gamma \Rightarrow \delta$.

Let R be a rule of $\mathrm{CHA}_<$ of the form

$$\mathrm{R}\frac{\Gamma_1 \Rightarrow \delta_1 \qquad \dots \qquad \Gamma_n \Rightarrow \delta_n}{\Gamma \Rightarrow \delta}$$

Fix $t \in \textsc{Term}(\Gamma \Rightarrow \delta)$ and $t' \in \textsc{Term}(\Gamma_i \Rightarrow \delta_i)$. The term t' is called a *precursor* of t, denoted $t' \leftarrow^i_R t$ if one of the following three conditions holds:

- R is an instance of (Sub) and $\Gamma = \Gamma'[s/x], \delta = \delta'[s/x]$ and $t = t'[s/x]$;
- or R is an instance of a rule *other than* (Sub) and $t = t'$;
- or R is an instance of (=L) and $\Gamma = \Gamma''[s/x, t/y], \Gamma' = \Gamma''[t/x, s/y]$ and analogously for the δ and there exists a term t'' such that $t = t''[s/x, t/y]$ and $t' = t''[t/x, s/y]$.

Now let Π be $\mathrm{CHA}_<$ pre-proof and let $(\Gamma_i \Rightarrow \delta_i)_{i\in\omega}$ be an infinite branch of Π. A trace along said branch is a sequence of terms $(t_i)_{i\in\omega}$ and an offset N such that for every $i \in \omega$ we have $t_i \in \textsc{Term}(\Gamma_{N+i} \Rightarrow \delta_{N+i})$ and furthermore, if $\Gamma_{N+i+1} \Rightarrow \delta_{N+i+1}$ is the jth premise of $\mathrm{CHA}_<$-rule R:

(i) either $t_{i+1} \leftarrow^j_R t_i$

(ii) or $s < t_i \in \Gamma$ and $t_{i+1} \leftarrow^j_R t_i$.

A trace is said to be progressing if case (ii) applies infinitely often along it. ⌟

Definition (**$\mathrm{CHA}_<$-proof**) A $\mathrm{CHA}_<$ pre-proof Π is a proof if every infinite branch of Π has a progressing trace. We write $\mathrm{CHA}_< \vdash \Gamma \Rightarrow \delta$ to mean that there exists a $\mathrm{CHA}_<$-proof Π with endsequent $\Gamma \Rightarrow \delta$. ⌟

2.3.2 Brotherston-Simpson Cyclic Arithmetic

2.3.3 Translating between Cyclic Arithmetics

2.4 Complexity results

Bibliography

Afshari, Bahareh and Graham E. Leigh. *Finitary Proof Systems for Kozen's ⊠*. Oberwolfach Preprint Series 2016–26. Mathematisches Forschungsinstitut Oberwolfach, 30th Dec. 2016.

– "Lyndon Interpolation for Modal ⊠-Calculus". In: *Language, Logic, and Computation*. Ed. by Aybüke Özgün and Yulia Zinova. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2022, pp. 197–213. DOI: 10.1007/978-3-030-98479-3_10.

Afshari, Bahareh, Graham E. Leigh and Guillermo Menéndez Turata. "Uniform Interpolation from Cyclic Proofs: The Case of Modal Mu-Calculus". In: *Automated Reasoning with Analytic Tableaux and Related Methods*. Vol. 12842. Cham: Springer International Publishing, 2021, pp. 335–353. DOI: 10.1007/978-3-030-86059-2_20.

– "A Cyclic Proof System for Full Computation Tree Logic". In: *31st EACSL Annual Conference on Computer Science Logic (CSL 2023)*. Vol. 252. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2023.

Afshari, Bahareh and Dominik Wehr. "Abstract Cyclic Proofs". In: *Logic, Language, Information, and Computation*. Ed. by Agata Ciabattoni, Elaine Pimentel and Ruy J. G. B. de Queiroz. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2022, pp. 309–325. DOI: 10.1007/978-3-031-15298-6_20.

Baelde, David, Amina Doumane and Alexis Saurin. "Infinitary Proof Theory: The Multiplicative Additive Case". In: *25th EACSL Annual Conference on Computer Science Logic (CSL 2016)*. Ed. by Jean-Marc Talbot and Laurent Regnier. Vol. 62. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016, 42:1–42:17. DOI: 10.4230/LIPIcs.CSL.2016.42.

Berardi, Stefano and Makoto Tatsuta. "Equivalence of Inductive Definitions and Cyclic Proofs under Arithmetic". In: *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. 2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS). June 2017, pp. 1–12. DOI: 10.1109/LICS.2017.8005114.

Berardi, Stefano and Makoto Tatsuta. “Equivalence of Intuitionistic Inductive Definitions and Intuitionistic Cyclic Proofs under Arithmetic”. 10th Dec. 2017. doi: 10.48550/arXiv.1712.03502.

Brotherston, James. “Sequent Calculus Proof Systems for Inductive Definitions”. PhD thesis. University of Edinburgh, Nov. 2006.

Brotherston, James, Dino Distefano and Rasmus L. Petersen. “Automated Cyclic Entailment Proofs in Separation Logic”. In: *Automated Deduction – CADE-23*. Ed. by Nikolaj Bjørner and Viorica Sofronie-Stokkermans. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2011, pp. 131–146. doi: 10.1007/978-3-642-22438-6_12.

Brotherston, James, Nikos Gorogiannis and Rasmus L. Petersen. “A Generic Cyclic Theorem Prover”. In: *Programming Languages and Systems*. Ed. by Ranjit Jhala and Atsushi Igarashi. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2012, pp. 350–367. doi: 10.1007/978-3-642-35182-2_25.

Das, Anupam and Damien Pous. “A Cut-Free Cyclic Proof System for Kleene Algebra”. In: *Automated Reasoning with Analytic Tableaux and Related Methods*. Ed. by Renate A. Schmidt and Cláudia Nalon. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2017, pp. 261–277. doi: 10.1007/978-3-319-66902-1_16.

Dax, Christian, Martin Hofmann and Martin Lange. “A Proof System for the Linear Time ⊠-Calculus”. In: *FSTTCS 2006: Foundations of Software Technology and Theoretical Computer Science*. Ed. by S. Arun-Kumar and Naveen Garg. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2006, pp. 273–284. doi: 10.1007/11944836_26.

Docherty, Simon and Reuben N. S. Rowe. “A Non-wellfounded, Labelled Proof System for Propositional Dynamic Logic”. In: *Automated Reasoning with Analytic Tableaux and Related Methods*. Ed. by Serenella Cerrito and Andrei Popescu. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 335–352. doi: 10.1007/978-3-030-29026-9_19.

Hazard, Emile and Denis Kuperberg. “Cyclic Proofs for Transfinite Expressions”. In: *30th EACSL Annual Conference on Computer Science Logic (CSL 2022)*. Ed. by Florin Manea and Alex Simpson. Vol. 216. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 23:1–23:18. doi: 10.4230/LIPIcs.CSL.2022.23.

Kori, Mayuko, Takeshi Tsukada and Naoki Kobayashi. “A Cyclic Proof System for HFLN”. In: *29th EACSL Annual Conference on Computer Sci-

ence Logic (CSL 2021). Ed. by Christel Baier and Jean Goubault-Larrecq. Vol. 183. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2021, 29:1–29:22. DOI: 10.4230/LIPIcs.CSL.2021.29.

Lee, Chin Soon, Neil D. Jones and Amir M. Ben-Amram. “The Size-Change Principle for Program Termination”. In: *Proceedings of the 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. POPL ’01. New York, NY, USA: Association for Computing Machinery, 1st Jan. 2001, pp. 81–92. DOI: 10.1145/360204.360210.

Marti, Johannes and Yde Venema. “A Focus System for the Alternation-Free ⊠-Calculus”. In: *Automated Reasoning with Analytic Tableaux and Related Methods*. Ed. by Anupam Das and Sara Negri. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021, pp. 371–388. DOI: 10.1007/978-3-030-86059-2_22.

Niwiński, Damian and Igor Walukiewicz. “Games for the ⊠-Calculus”. In: *Theoretical Computer Science* 163.1 (30th Aug. 1996), pp. 99–116. DOI: 10.1016/0304-3975(95)00136-0.

Nollet, Rémi, Alexis Saurin and Christine Tasson. “PSPACE-Completeness of a Thread Criterion for Circular Proofs in Linear Logic with Least and Greatest Fixed Points”. In: *Automated Reasoning with Analytic Tableaux and Related Methods*. Ed. by Serenella Cerrito and Andrei Popescu. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 317–334. DOI: 10.1007/978-3-030-29026-9_18.

Ong, C.-H.L. “On Model-Checking Trees Generated by Higher-Order Recursion Schemes”. In: *21st Annual IEEE Symposium on Logic in Computer Science (LICS’06)*. 21st Annual IEEE Symposium on Logic in Computer Science (LICS’06). Aug. 2006, pp. 81–90. DOI: 10.1109/LICS.2006.38.

Savateev, Yury and Daniyar Shamkanov. “Non-Well-Founded Proofs for the Grzegorczyk Modal Logic”. In: *The Review of Symbolic Logic* 14.1 (1 Mar. 2021), pp. 22–50. DOI: 10.1017/S1755020319000510.

Shamkanov, Daniyar. “Circular Proofs for the Gödel-Löb Provability Logic”. In: *Mathematical Notes* 96.3 (3 1st Sept. 2014), pp. 575–585. DOI: 10.1134/S0001434614090326.

Simpson, Alex. “Cyclic Arithmetic Is Equivalent to Peano Arithmetic”. In: *Foundations of Software Science and Computation Structures*. Ed. by Javier Esparza and Andrzej S. Murawski. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2017, pp. 283–300. DOI: 10.1007/978-3-662-54458-7_17.

Sprenger, Christoph and Mads Dam. “On Global Induction Mechanisms in a ⊠-Calculus with Explicit Approximations”. In: *RAIRO - Theoretical Informatics and Applications* 37.4 (Oct. 2003), pp. 365–391. DOI: 10.1051/ita:2003024.

Tellez, Gadi and James Brotherston. “Automatically Verifying Temporal Properties of Pointer Programs with Cyclic Proof”. In: *Automated Deduction – CADE 26*. Ed. by Leonardo de Moura. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2017, pp. 491–508. DOI: 10.1007/978-3-319-63046-5_30.

Wehr, Dominik. “An Abstract Framework for the Analysis of Cyclic Derivations”. MSc thesis. Amsterdam, Netherlands: University of Amsterdam, 26th Aug. 2021.

Sammanfattning