On Natural Deduction in First-Order Fixpoint Logics*

Andrzej SZAŁAS†
Institute of Informatics, University of Warsaw
02-097 Warsaw, ul. Banacha 2, Poland
e-mail: szalas@mimuw.edu.pl

Abstract

In the current paper we present a powerful technique of obtaining natural deduction proof systems for first-order fixpoint logics. The term *fixpoint logics* refers collectively to a class of logics consisting of modal logics with modalities definable at meta-level by fixpoint equations on formulas. The class was found very interesting as it contains most logics of programs with e.g. dynamic logic, temporal logic and the μ-calculus among them.

In this paper we present a technique that allows us to derive automatically natural deduction systems for modal logics from fixpoint equations defining the modalities.

1 Introduction

A great deal of attention has been devoted to formalisms dealing with fixpoints. Denotational semantics, domain theory, complexity theory, specification languages - these are just a few computer science examples of such formalisms. As logic was widely applied in various areas of computer science, those formalisms have their natural counterparts in calculus, which we call *fixpoint calculus*, or *fixpoint logics*.

The current paper is devoted to axiomatizing a large class of multimodal logics with modalities definable by least fixpoints of equations on formulas. The approach we consider is discussed in [10,11], where both complete and relatively complete Hilbert-like proof systems for the logics are given. The approach we investigate is close to that of μ-calculus (cf. e.g. [6]). The differences between those approaches are precisely discussed in [11]. Let us only recall that the most important differences are:

- we require a stronger assumptionon functionals defining the meaning of formulas, than the monotonicity that is required in the μ-calculus. Moreover, we do not deal with greatest fixpoints, but the least ones only (see, however, discussion provided in section 6)

*Published in *Fundamenta Informaticae*, 26, 1, 1996, 81–94
†This research was supported in part by the KBN grant 3 P406 019 06

- we do not assume any particular language of considered logics, while the usual modal versions of the μ-calculus inherit the whole background of the underlying logic. We find this feature a disadvantage of those approaches.

As mentioned earlier, Hilbert-like proof systems for considered logics are presented in [10,11]. On the other hand, there is also another important method of defining proof systems, so called natural deduction method for the first time proposed independently by G. Gentzen and S. Jaśkowski. It is worth emphasizing here that Hilbert style of presentation of proof systems is more suitable for humans, while natural deduction can be much easier automated (cf. e.g. [3]). In what follows we shall present both complete and relatively complete natural deduction systems for fixpoint logics. As the logics we consider are usually totally undecidable, they cannot be completely axiomatized by effective proof systems. However, the systems we present can be implemented directly, or at least suggest possible implementations. The non-effective parts of the systems can be replaced by finite formal systems of arithmetics (cf. e.g. [12]). The obtained implementations are not as strong as the initial proof systems. However, taking sufficiently strong finitistic formal systems of the second-order arithmetics, one can obtain quite powerful systems. The techniques of measuring the strength of such implementations can be found in the literature (cf. e.g. [1]).

2 Preliminary notions

Let us first establish a logical framework assumed in this paper. The logics we consider are extensions of classical first-order logic. By M we shall denote an enumerable set of nonclassical connectives (or, in other words, modalities). For the sake of simplicity we assume that the connectives are unary. The presented approach can easily be extended to nonclassical connectives that have more than one argument (cf. [10] and also example 2.3). In the sequel we shall always assume that a first-order signature is fixed. By L we shall then denote the set of many-sorted classical first-order formulas.

Definition 2.1 Let M be an enumerable set of nonclassical connectives. We form an *M-extension of classical first-order logic*, M-logic in short, as triple $\mathcal{L} = \langle L(M), \mathcal{C}, \models \rangle$, where:

1. $L(M)$ is the set of formulas obtained from L augmented with the following syntax rule:
   - for any $m \in M$ and $A \in L(M)$, $m(A) \in L(M)$
2. $\mathcal{C}$ is a class of admissible interpretations (we assume that $\mathcal{C}$ is a subclass of classical first-order interpretations in relational structures)
3. $\models$ is a satisfiability relation that agrees with the classical one for classical first-order formulas (for $\mathcal{M} \in \mathcal{C}, A \in L(M)$ and valuation v of free variables, $\mathcal{M}, v \models A$ means that A is satisfied by interpretation $\mathcal{M}$ and valuation v). □

In what follows we shall define the notion of fixpoint logics, as understood in this paper. First, however, let us consider two examples that illustrate the main idea (cf. also [10,11]).

Example 2.2 Let $\langle P^* \rangle A$ be a modality of dynamic logic (cf. e.g. [4]) meaning that there is a nondeterministic iteration of program P, with results satisfying the formula A, i.e.

$$\mathcal{M}, v \models_{DL} \langle P^* \rangle A \text{ iff there is } i \in \omega \text{ such that } \mathcal{M}, v \models_{DL} \langle P \rangle^i A,$$

where by $\langle P \rangle^i$ we mean $\langle P \rangle$ repeated i-times and $\models_{DL}$ denotes the satisfiability relation of the dynamic logic. Then for all $\mathcal{M}$ and v,

- $\mathcal{M}, v \models_{DL} \langle P^* \rangle A \;\leftrightarrow\; A \vee \langle P \rangle \langle P^* \rangle A$
- $\mathcal{M}, v \models_{DL} \langle P^* \rangle A$ iff there is $i \in \omega$ such that $\mathcal{M}, v \models_{DL} G^i_{\langle P^* \rangle A}(\mathbf{false})$, where $G_{\langle P^* \rangle A}(x) = A \vee \langle P \rangle x$.

Observe that the first of the above propositions can be reformulated as follows:

- $\mathcal{M}, v \models_{DL} \langle P^* \rangle A \;\leftrightarrow\; G_{\langle P^* \rangle A}(\langle P^* \rangle A)$. □

In the following example we consider **atnext** operator of temporal logic. The original operator is a two-argument one. Since we deal with unary modalities only, we introduce infinitely many operators $\mathbf{atnext}_B$, where B's are temporal formulas.

Example 2.3 Let $A\mathbf{atnext}_B$ be a modality of linear time temporal logic (cf. e.g. [7]) meaning that there is a future time point satisfying formula B and in the first such a point formula A is satisfied, i.e.

$$\mathcal{M}, v \models_{TL} A\mathbf{atnext}_B \text{ iff there is } i \in \omega - \{0\} \text{ such that } \mathcal{M}, v \models_{TL} \bigcirc^i (A \wedge B),$$
$$\text{and for all } 0 < j < i,\ \mathcal{M}, v \models_{TL} \bigcirc^j (\neg B),$$

where by $\bigcirc^k$ we mean $\bigcirc$ repeated k-times, and $\models_{TL}$ denotes the satisfiability relation of temporal logic. Then for all $\mathcal{M}$ and v,

- $\mathcal{M}, v \models_{TL} A\mathbf{atnext}_B \;\leftrightarrow\; \bigcirc(A \wedge B) \vee \bigcirc(\neg B \wedge A\mathbf{atnext}_B)$
- $\mathcal{M}, v \models_{TL} A\mathbf{atnext}_B$ iff there is $i \in \omega$ such that $\mathcal{M}, v \models_{TL} G^i_{A\mathbf{atnext}_B}(\mathbf{false})$, where $G_{A\mathbf{atnext}_B}(x) = \bigcirc(A \wedge B) \vee \bigcirc(\neg B \wedge x)$.

Note that the first of the above propositions can be reformulated as follows:

- $\mathcal{M}, v \models_{TL} A\mathbf{atnext}_B \;\leftrightarrow\; G_{A\mathbf{atnext}_B}(A\mathbf{atnext}_B)$. □

The above examples show the most essential characterization of nonclassical connectives in considered logics. Namely, equivalences given above have the following common form:

$$x \leftrightarrow G(x).$$

Moreover, each of the defined modalities is characterized as least upper bound of the set $\{G^i(\mathbf{false}) \;:\; i \in \omega\}$ of formulas.

Let us now provide a more precise definition of definability of nonclassical connectives by means of fixpoint equations (where equality on formulas is interpreted as usual

equivalence). Note that in the below definition we require some additional well-founded relation on nonclassical connectives. That is a bit technical point in the definition. However, the required relation can usually be found in a natural way. In what follows we shall then assume that the set of nonclassical connectives, M, is always supplemented by a well-founded relation $<_M$.

Definition 2.4 We say that set of formulas $\mathbf{G}(M) = \{G_{m(A)} : m \in M, A \in L(M)\}$ *defines set* M of nonclassical connectives of M-logic $\mathcal{L}$ provided that the following conditions hold:

1. for any interpretation $\mathcal{M}$ of $\mathcal{L}$ and valuation v of free variables,
   (a) $\mathcal{M}, v \models m(A) \leftrightarrow G_{m(A)}(m(A))$
   (b) $\mathcal{M}, v \models m(A)$ iff there is $i \in \omega$ such that $\mathcal{M}, v \models G^i_{m(A)}(\mathbf{false})$
2. there is a well-founded relation $<_M$ on M such that righthand sides of equivalences defining functionals $G_{m(A)}$ contain (syntactically) only connectives less (w.r.t. $<_M$) than m.[1] □

For examples of functionals defining various logics see [11]. Let us now define a notion of monotonicity that plays a key rôle in this paper. It is worth mentioning here that we use two kinds of arrows, $\Rightarrow$ and $\rightarrow$. The first one separates the two parts of a sequent and the second one stands for the usual implication.

Definition 2.5

1. Given an M-logic, we shall say that set M of nonclassical connectives is *monotone* iff for any interpretation $\mathcal{M}$, nonclassical connective $m \in M$, and formulas A, B:
$$\mathcal{M} \models A \rightarrow B \text{ implies } \mathcal{M} \models m(A) \rightarrow m(B)$$
2. Given an M-logic, we shall say that a functional G is monotone iff for any interpretation $\mathcal{M}$ and formulas A, B:
$$\mathcal{M} \models A \rightarrow B \text{ implies } \mathcal{M} \models G(A) \rightarrow G(B)$$
3. We say that an M-logic is *monotone* iff M is monotone and there is a set of monotone functionals defining connectives of M.

To indicate the fact that set of nonclassical connectives of monotone M-logic is definable by a set $\mathbf{G}(M)$ of monotone functionals we shall write $(M, \mathbf{G})$-logic instead of M-logic. □

Observe that the condition 1(b) of definition 2.4 holds whenever we deal with the continuous functionals. It is also implied by monotonicity and partial computability of functionals (cf. [5]).

What now remains to define is the natural deduction method.

[1] As observed by the referee, it suffices to assume that in $G_{m(A)}$ can appear only smaller connectives or arbitrary connectives, but with smaller formulas

Definition 2.6 Let $\mathcal{L}$ be an M-logic.

1. By a *sequent* of logic $\mathcal{L}$ we shall mean any expression of the form $\Gamma \Rightarrow \Delta$, where both Γ and Δ are finite sequences[2] of formulas of $\mathcal{L}$.
2. By $\mathcal{M}, v \models \Gamma \Rightarrow \Delta$ we shall mean that $\mathcal{M}, v \models \bigwedge_{A \in \Gamma} A \rightarrow \bigvee_{A \in \Delta} A$.
3. By a natural deduction proof system we shall mean any pair $\langle Ax, R \rangle$ such that
   (a) Ax, called the set of *axioms*, is any set of sequents of $\mathcal{L}$
   (b) R is any set of derivation rules of the form $\Sigma \vdash S$, where Σ is a set of sequents of $\mathcal{L}$, and S is a sequent of $\mathcal{L}$.
4. We say that sequent S is *indecomposable* in a given natural deduction proof system iff it is an axiom or no rule of the system is applicable to S. A sequent is *decomposable* iff it is not indecomposable. □

Definition 2.7 Let $P = \langle Ax, R \rangle$ be a natural deduction system for logic $\mathcal{L}$.

1. By a decomposition tree of a sequent S in proof system P we shall mean a rooted tree with nodes labelled by sequents, such that
   (a) the root of the tree is labelled by S
   (b) all leaves of the tree are labelled by indecomposable sequents
   (c) any node n in the tree is either labelled by an element of Ax, or by sequent S for which there is a derivation rule $\mathbf{S} \vdash S$ in R such that
      i. $\mathbf{S} = \{t : \text{t is a label of a son of } n \text{ in the tree}\}$
      ii. the first decomposable formula (counting from left to right) of sequent labelling n is decomposed.
2. By a proof of sequent S in P we shall mean a decomposition tree of S satisfying the following additional conditions
   (a) the height of the tree is finite
   (b) all leaves are labelled by axioms of Ax. □

Note that proofs are carried out top down and afterwards read bottom up. Note also that, according to notational conventions used in the literature, by $\Gamma, \Delta, \Pi, \Sigma$ we shall denote finite sets of formulas. Similarly, by Γ, A, Δ we shall mean set $\Gamma \cup \{A\} \cup \Delta$. Thus colon corresponds to set-theoretical union. Semicolon is used to separate sequents from each other.

It is worth emphasizing here that both proof systems we present are cut-free. This means that the *cut* rule is not included in those proof system. This, of course, considerably simplifies both the search for proofs and possible implementations of the proof systems. (In fact, a weak form of *cut* rule appears in definition 4.1 (rule $5'(a)$). This, however, as we shall see, causes no further implementation problems.)

[2]It is sometimes convenient to consider sets instead of sequences. We shall sometimes use this convention, too.

3 An infinitary proof system

Let us now define an infinitary proof systems for fixpoint logics. For the sake of simplicity, in the classical part of the proof system we introduce rules for $\neg$, $\wedge$ and $\forall$ only. Other boolean connectives and the existential quantifier $\exists$ can be defined by the above ones as usually. The corresponding rules can easily be derived.

Observe that a natural deduction proof system for $L_{\omega_1\omega}$ that could be used here is given in [8]. The one we define is adapted to the formalism we deal with.

In what follows we shall always assume that an enumeration of the set of terms is given. By t_i, where $i \in \omega$, we shall then denote the i-th term (w.r.t. the enumeration). For a sequent S labelling node, say n, in a decomposition tree T, by Γ_n^S (or Δ_n^S) we shall mean $\bigcup_{l\in N}\Gamma_l$ (or $\bigcup_{l\in N}\Delta_l$, respectively), where N is the set of all nodes on the path from n to the root of the tree (including n) and Γ_l, Δ_l denote respective parts of sequent labelling node l. In what follows we often write Γ^S and Δ^S instead of Γ_n^S and Δ_n^S.

Definition 3.1 Let $\mathcal{L}$ be an $(M, \mathbf{G})$-logic. By $IP_{\mathcal{L}}$ we shall mean the following proof system

I. axioms:
$\vdash \Gamma \Rightarrow \Delta$, when $\Gamma \cap \Delta \neq \emptyset$

II. rules:

1. (a) $A, \Gamma \Rightarrow \Sigma, \Delta \ \vdash \ \Gamma \Rightarrow \Sigma, \neg A, \Delta$
   (b) $\Sigma, \Gamma \Rightarrow A, \Delta \ \vdash \ \Sigma, \neg A, \Gamma \Rightarrow \Delta$
2. (a) $\Sigma, A, B, \Gamma \Rightarrow \Delta \ \vdash \ \Sigma, A \wedge B, \Gamma \Rightarrow \Delta$
   (b) $\Gamma \Rightarrow \Sigma, A, \Delta$; $\Gamma \Rightarrow \Sigma, B, \Delta \ \vdash \ \Gamma \Rightarrow \Sigma, A \wedge B, \Delta$
3. (a) $\Sigma, A(x \leftarrow t), \Gamma \Rightarrow \Delta, \neg\forall x(A(x)) \vdash \ \Sigma, \forall x(A(x)), \Gamma \Rightarrow \Delta$,
   where t is the first term (w.r.t. given enumeration) for which $A(x \leftarrow t)$ does not appear in $\Gamma^{\Sigma, \forall x(A(x)), \Gamma \Rightarrow \Delta}$, and $A(x \leftarrow t)$ denotes the formula obtained from A by replacing x by t with renaming the free variables of t which are bound in A, if necessary
   (b) $\Gamma \Rightarrow \Sigma, A(x), \Delta \ \vdash \ \Gamma \Rightarrow \Sigma, \forall x(A(x)), \Delta$,
   where variable x does not appear neither in Γ, nor in Δ
4. for all $m \in M$ and formula A such that $G_{m(A)}(x)$ is a constant functional (syntactically, i.e. functional containing no occurrences of x) we assume the following rules:
   (a) $\Sigma, G_{m(A)}(\mathbf{false}), \Gamma \Rightarrow \Delta \ \vdash \ \Sigma, m(A), \Gamma \Rightarrow \Delta$
   (b) $\Gamma \Rightarrow \Sigma, G_{m(A)}(\mathbf{false}), \Delta \ \vdash \ \Gamma \Rightarrow \Sigma, m(A), \Delta$
5. for all $m \in M$ other than those above we assume the following rules:
   (a) $\Gamma \Rightarrow \Sigma, G^i_{m(A)}(\mathbf{false}), \Delta, m(A) \ \vdash \ \Gamma \Rightarrow \Sigma, m(A), \Delta$,
   where i is the smallest natural number for which $G^i_{m(A)}(\mathbf{false})$ does not appear in $\Delta^{\Gamma \Rightarrow \Sigma, m(A), \Delta}$
   (b) $\{\Sigma, G^i_{m(A)}(\mathbf{false}), \Gamma \Rightarrow \Delta\}_{i\in\omega} \ \vdash \ \Sigma, m(A), \Gamma \Rightarrow \Delta$. □

Note that the rules $4(a)$ and $4(b)$ are special cases of rules $5(a)$ and $5(b)$. We introduced them in order to simplify the obtained proof systems. Constant functionals appear in considered logic rather frequently but, on the other hand, need no infinitary characterization.

One can find some context conditions, referring to the path in decomposition tree (cf. rules $3(a)$ and $5(a)$), somewhat unusual. We introduced them to be closer to implementation of given proof systems. On can, however, reformulate them into the form of "pure" natural deduction method as follows:

$\underline{3(a)}$ $\Gamma, A(x \leftarrow t_{i+1}), \Sigma \Rightarrow \Delta, \neg\forall x(A(x)), \neg A(x \leftarrow t_0), ..., \neg A(x \leftarrow t_i), \neg A(x \leftarrow t_{i+1})$
$\vdash \Gamma, \forall x(A(x)), \Sigma \Rightarrow \Delta, \neg A(x \leftarrow t_0), ..., \neg A(x \leftarrow t_i),$

$\underline{5(a)}$ $\Gamma \Rightarrow \Delta, G^i_{m(A)}(\mathbf{false}), \Sigma, m(A), G^i_{m(A)}(\mathbf{false}) \vdash \Gamma \Rightarrow \Delta, m(A), \Sigma,$
where i is the smallest natural number for which $G^i_{m(A)}(\mathbf{false})$ does not appear in $\Delta \cup \Sigma$.

Observe that both $A(x \leftarrow t_{i+1})$ and $G^i_{m(A)}(\mathbf{false})$ appear in the premises twice, for the second time artificially, in a context where they can never be decomposed, since we only allow the first decomposable formula to be decomposed (cf. definition 2.7). In our case, always $\neg\forall x(A(x))$ or $m(A)$ is then to be decomposed before the latter occurrence of $\neg A(x \leftarrow t)$ or $G^i_{m(A)}(\mathbf{false})$, respectively. That is a technical trick, due to which all suitable formulas remain all the time inside of sequents so that, in a sense, a sequent remembers which formulas were used during its proof. This makes it possible to check context conditions which are directly related to a sequent (but indirectly, of course, again to the whole path which is now stored inside of the sequent).

Note also, that formula $\neg\forall x(A(x))$ appears at righthand sides of sequent in the premises of both $3(a)$ and $\underline{3(a)}$. This is again a technical trick, due to which sequents remember that formula $\forall x(A(x))$ can still be decomposed, but after the decomposition of formulas in Δ. (That is, of course, not necessary in case of rules $5(a)$ and $\underline{5(a)}$, as respective formulas already appear at the rightmost sides of sequents.)

Definition 3.2 Let P be a proof system for the logic $\mathcal{L} = \langle F, \mathcal{C}, \models \rangle$. Then

1. we shall say that proof system P is *sound* iff for any sequent $\Gamma \Rightarrow \Delta$ provable in P, $\models \Gamma \Rightarrow \Delta$
2. we shall say that proof system P is *complete* iff any sequent $\Gamma \Rightarrow \Delta$ such that $\models \Gamma \Rightarrow \Delta$ is provable in P. □

The following theorem provides us with a characterization of proof system $IP_{\mathcal{L}}$. Observe that the proof follows from that given by Lopez-Escobar [8] for $L_{\omega_1\omega}$.

Theorem 3.3 For any $(M, \mathbf{G})$-logic $\mathcal{L}$, proof system $IP_{\mathcal{L}}$ is sound and complete. □

4 A relatively complete proof system

In this section we define proof systems $RP_{\mathcal{L}}$ which are obtained from the previous one by replacing infinitary proof rules. In what follows we shall always assume, that the

first-order signature contains (at least) constant symbols 0 and 1, two binary function symbols + and $*$, and a binary relation symbol $\leq$.

Definition 4.1 Let $\mathcal{L}$ be an $(M, \mathbf{G})$-logic. By $RP_{\mathcal{L}}$ we shall mean the proof system obtained from the infinitary system $IP_{\mathcal{L}}$ (cf. definition 3.1) by replacing the proof rule 5 by the following ones:

5'. (a) $G_{m(A)}(C) \Rightarrow C;\ \Gamma, C, \Sigma \Rightarrow \Delta \ \vdash\ \Gamma, m(A), \Sigma \Rightarrow \Delta$

(b) $C(n \leftarrow n+1) \Rightarrow G_{m(A)}(C(n));\ C(n \leftarrow 0) \Rightarrow \emptyset;\ \Gamma \Rightarrow \Sigma, \exists n(C(n)), \Delta \vdash\ \Gamma \Rightarrow \Sigma, m(A), \Delta,$

where n does not appear in $m(A)$. □

Note that the presence of formula C in rules $5'(a)$ and $5'(b)$ seems to complicate the search for proofs or even make it impossible to automatize. However, as it will follow from the proof of theorem 4.5, the search for a suitable formula can also in this case be automated. The formula obtained from the proof, as a general one, is usually not the simplest one. Some heuristics are then necessary to make the process of proving theorems more efficient.

Let us now discuss the notion of relative completeness. It was for the first time considered by Cook (cf. [2]) in context of Hoare logics. Cook separated the reasoning about programs from reasoning about properties of data structures. He then restricted the class of admissible interpretations to so called *expressive* interpretations only. Later it turned out, that one has to restrict himself to *arithmetical* interpretations when considering logics more expressive than that of Hoare. Arithmetical completeness, reflecting this restriction, has then been derived from relative completeness by Harel in [4], in context of dynamic logic. Harel gave finitary proof rules for first-order dynamic logic that allow us to eliminate programs from formulas of the logic. As first-order dynamic logic is totally undecidable, there was of course price to pay, namely the set of axioms forms now a totally undecidable set. On the other hand, those axioms, as classical first-order properties of data structures are supposed to be known by a programmer, who should never write programs based on unknown properties of data. Yet another restriction of class of interpretations was considered in [10], where the only admissible interpretations are *strictly arithmetical* interpretations. Such a class of interpretations is a proper subclass of arithmetical interpretation. It is, however, still large and worth interest. For instance, domains of finite stacks, queues, trees, arrays, symbols etc. with usual operations on them are all strictly arithmetical. More precise definition follows.

Definition 4.2 Let $\mathcal{L} = \langle F, \mathcal{C}, \models \rangle$ be an M-logic. Interpretation $\mathcal{M} \in \mathcal{C}$ is called *strictly arithmetical* (*s-arithmetical*, in short) provided that:

1. $\mathcal{M}$ contains sort ω of natural numbers together with constants 0, 1, functions +, $*$ and relation $\leq$ (interpreted as usual)
2. for each sort s of $\mathcal{M}$ there is an effective binary relation e_s encoding elements of sort s, i.e. such that for each x of sort s there is exactly one $i \in \omega$ with $e_s(x, i)$ true in M. □

We are now ready to define notions of relative and strictly arithmetical soundness and completeness.

Definition 4.3 Let P be a proof system for the logic $\mathcal{L} = \langle F, \mathcal{C}, \models \rangle$. Then

1. we say that P is *sound (complete) for $\mathcal{L}$ relative to class $\mathcal{I} \subseteq \mathcal{C}$* provided that for any interpretation $\mathcal{M} \in \mathcal{I}$ and any sequent $\Gamma \Rightarrow \Delta$ of $\mathcal{L}$,

$$\vdash_{Th_{\mathcal{M}}} \Gamma \Rightarrow \Delta \text{ implies (is implied by) } \mathcal{M} \models \Gamma \Rightarrow \Delta,$$

where $Th_{\mathcal{M}}$ denotes the first-order theory of interpretation $\mathcal{M}$, i.e. the set $\{A \in L \ : \ \mathcal{M} \models A\}$, and $\vdash_{Th_{\mathcal{M}}}$ denotes the syntactic consequence relation of proof system P augmented with the following set of axioms:

$$\{\Pi \Rightarrow \Sigma_1, A, \Sigma_2 \ : \ A \in Th_{\mathcal{M}}\}$$

2. we say that P is *s-arithmetically sound (complete)* provided that it is sound (complete) for $\mathcal{L}$ relative to the class of s-arithmetical interpretations. □

In order to simplify our considerations, in what follows we shall consider one-sorted s-arithmetical interpretations with sort ω, operations $0, 1, +, *$ and additional functions of signature $\omega \to \omega$. In the presence of encoding relations this can be done without loss of generality. Namely, functions and relations on sorts other than ω can be represented by functions with signature $\omega \to \omega$ or $\omega \to \{0, 1\}$, respectively.

Let us now briefly discuss the notion of partial recursive functional, as it is needed in the proof of s-arithmetical completeness of proof system $RP_{\mathcal{L}}$. Namely, by a partial recursive functional we shall mean any functional that, interpreted in s-arithmetical interpretation, is partial recursive (perhaps relative to some oracle). That is, to say, a functional G is partial recursive whenever for each formula A and vector of variables $\mathbf{x}$, given an oracle answering whether $A(\mathbf{x})$ is true, the question whether $G(A)(\mathbf{x})$ is true, is partial recursive. This notion of partial recursiveness is well known and its precise definition need not be quoted here. The definition of partial recursive functionals that perhaps best serves our purposes is to be found in the book [5].

Now we are ready to prove the main results of this section.

Theorem 4.4 For any $(M, \mathbf{G})$-logic $\mathcal{L}$, proof system $RP_{\mathcal{L}}$ is s-arithmetically sound.

Proof

Soundness of rules $1 - 4$ easily follows from their soundness in proof system $IP_{\mathcal{L}}$. Let us then first prove soundness of rule $5'(a)$.

Assume that the premises are true in some interpretation $\mathcal{M}$. We shall then show that for all $i \in \omega$,

$$\mathcal{M} \models \Gamma, G^i_{m(A)}(\mathbf{false}), \Sigma \Rightarrow \Delta.$$

Let us first show that for all $i \in \omega$,

$$\mathcal{M} \models G^i_{m(A)}(\mathbf{false}) \Rightarrow C.$$

We proceed by induction on i. The case of $i = 0$ is trivial, for $G^0_{m(A)}$ applied to any formula, is, by convention, **false**. Assume that our claim is true for some $i \in \omega$. We shall show that it then remains true also for $i + 1$. Note that, by inductive assumption and monotonicity of G (cf. definition 2.5), we have that $\mathcal{M} \models G^i_{m(A)}(\mathbf{false}) \Rightarrow C$ implies

$\mathcal{M} \models G(G^i_{m(A)}(\mathbf{false})) \Rightarrow G_{m(A)}(C)$. Since the first of premises, $G_{m(A)}(C) \Rightarrow C$, is assumed valid in $\mathcal{M}$, we also have that $\mathcal{M} \models G(G^i_{m(A)}(\mathbf{false})) \Rightarrow C$, i.e. $\mathcal{M} \models G^{i+1}_{m(A)}(\mathbf{false}) \Rightarrow C$, which completes the proof of our claim.

By the second of premises of the rule we now have that for all $i \in \omega$,

$$\mathcal{M} \models \Gamma, G^i_{m(A)}(\mathbf{false}), \Sigma \Rightarrow \Delta.$$

Now the rest of the proof of soundness of rule $5'(a)$ can be carried out just like in the case of rule $5(b)$ of proof system $IP_{\mathcal{L}}$.

What now remains to prove is the soundness of rule $5'(b)$. Assume that all premises of rule $5'(b)$ are true in interpretation $\mathcal{M}$. We shall show that for all $i \in \omega$,

$$\mathcal{M} \models C(i) \Rightarrow G^i_{m(A)}(\mathbf{false}).$$

We proceed by induction on i. The case of $i = 0$. is trivial for formula $C(0) \Rightarrow G^0_{m(A)}(\mathbf{false})$ is just the second of premises. Assume $\mathcal{M} \models C(i) \Rightarrow G^i_{m(A)}(\mathbf{false})$. Then, by monotonicity of $G_{m(A)}$, $\mathcal{M} \models G_{m(A)}(C(i)) \Rightarrow G_{m(A)}(G^i_{m(A)}(\mathbf{false}))$. From the first premise of our rule we have $\mathcal{M} \models C(i+1) \Rightarrow G_{m(A)}(C(i))$. Thus $\mathcal{M} \models C(i+1) \Rightarrow G_{m(A)}(G^i_{m(A)}(\mathbf{false}))$, i.e. $\mathcal{M} \models C(i+1) \Rightarrow G^{i+1}_{m(A)}(\mathbf{false})$.

Now note that, by the third premise of our rule, $\mathcal{M} \models \Gamma \Rightarrow \Sigma, \exists n(C(n)), \Delta$. Thus, by the above and definition 2.4, $\mathcal{M} \models \Gamma \Rightarrow \Sigma, m(A), \Delta$, which proves the result. □

Theorem 4.5 For any $(M, \mathbf{G})$-logic $\mathcal{L}$, if all functionals of $\mathbf{G}$ are partial recursive then proof system $RP_{\mathcal{L}}$ is s-arithmetically complete.

Proof

The proof of s-arithmetical completeness can easily be reduced to the proof of similar theorem shown in [10] for Hilbert-like proof systems. One can use lemma 4.4 of [10], where the reasoning about fixpoint formulas is reduced to reasoning about classical first-order ones. Namely, formula C required in rule $5'$ is constructed there and appears to be a classical first-order one. Formula C that satisfies premises of rules $5'(a)$ and $5'(b)$ can defined inductively as follows:

$C(n \leftarrow 0) \leftrightarrow \mathbf{false}$

$C(n \leftarrow n+1) \leftrightarrow G_{m(A)}(C(n))$.

(Note that $\mathcal{M} \models m(A) \leftrightarrow \exists n(C(n)))$.

The only (and, in fact, most difficult) problem to be solved is that we still have to eliminate the inductive definition of C and find a formula that explicitly defines C. There is, however, a theorem in recursion theory (cf. e.g. theorem 3.5 in [5], p. 92) that guarantees that such an elimination is indeed possible (cf. also [10]). Moreover, the required formula can be constructed automatically.

Having a procedure of finding suitable formula C one can step by step eliminate nonclassical operators from sequents. As there can be only finitely many such operators, and $G_{m(A)}(C)$ can contain only new operators that are less than m (w.r.t. ordering $<_M$ required in definition 2.4), such an elimination terminates after finitely many steps. Those steps are reflected by application of suitable parts of rule 5.

This completes the proof of s-arithmetical completeness of $RP_{\mathcal{L}}$. □

The above theorems give us the following important characterization of proof system $RP_{\mathcal{L}}$, where $\mathcal{L}$ is an $(M, \mathbf{G})$-logic with all functionals of $\mathbf{G}$ partial recursive:

> if interpretation $\mathcal{M}$ is strictly arithmetical, then the set of sequents provable in $RP_{\mathcal{L}}$ augmented with set $\{\Pi \Rightarrow \Sigma_1, A, \Sigma_2 \ : \ A \in Th_{\mathcal{M}}\}$ of axioms is equal to the set of all sequents $\Gamma \Rightarrow \Delta$ for which $\mathcal{M} \models \Gamma \Rightarrow \Delta$. In particular, the set of all formulas A, for which sequent $\emptyset \Rightarrow A$ is provable in $RP_{\mathcal{L}}$ augmented with $\{\Pi \Rightarrow \Sigma_1, A, \Sigma_2 \ : \ A \in Th_{\mathcal{M}}\}$ is equal to the set of all formulas valid in interpretation $\mathcal{M}$.

5 Examples of applications

Let us now show two examples of application of the theorems given in the previous section.

Example 5.1 Consider modality $\langle P^* \rangle A$ of dynamic logic (cf. example 2.2). The following axioms and proof rules (together with some other ones for other modalities of dynamic logic, that can be easily derived from equations given e.g. in [10]) give sound and complete characterization of $\langle P^* \rangle A$.

- axioms and rules $1 - 3$ given in definition 3.1
- $\Gamma \Rightarrow \Sigma, G^i_{\langle P^* \rangle A}(\mathbf{false}), \Delta, \langle P^* \rangle A \ \vdash \ \Gamma \Rightarrow \Sigma, \langle P^* \rangle A, \Delta$,
  where i is the smallest natural number for which $G^i_{\langle P^* \rangle A}(\mathbf{false})$ does not appear in
  $\Delta^{\Gamma \Rightarrow \Sigma, \langle P^* \rangle A, \Delta}$
- $\{\Sigma, G^i_{\langle P^* \rangle A}(\mathbf{false}), \Gamma \Rightarrow \Delta\}_{i \in \omega} \ \vdash \ \Sigma, \langle P^* \rangle A, \Gamma \Rightarrow \Delta$.

After applying $G^i_{\langle P^* \rangle A}$ and making minor cosmetics, one can formulate the last two rules as follows:

- $\Gamma \Rightarrow \Sigma, \langle P \rangle^i A, \Delta, \langle P^* \rangle A \ \vdash \ \Gamma \Rightarrow \Sigma, \langle P^* \rangle A, \Delta$,
  where i is the smallest natural number for which $G^i_{\langle P^* \rangle A}(\mathbf{false})$ does not appear in
  $\Delta^{\Gamma \Rightarrow \Sigma, \langle P^* \rangle A, \Delta}$
- $\{\Sigma, \langle P \rangle^i A, \Gamma \Rightarrow \Delta\}_{i \in \omega} \ \vdash \ \Sigma, \langle P^* \rangle A, \Gamma \Rightarrow \Delta$.

After replacing the last two rules by

- $A \vee \langle P \rangle C \Rightarrow C; \ \Gamma, C, \Sigma \Rightarrow \Delta \ \vdash \ \Gamma, \langle P^* \rangle A, \Sigma \Rightarrow \Delta$
- $C(n \leftarrow n+1) \Rightarrow A \vee \langle P \rangle C(n); \ C(n \leftarrow 0) \Rightarrow \emptyset; \ \Gamma \Rightarrow \Sigma, \exists n(C(n)), \Delta \vdash \Gamma \Rightarrow \Sigma, \langle P^* \rangle A, \Delta$,
  where n does not appear in $\langle P^* \rangle A$

one obtains s-arithmetically sound and complete characterization of $\langle P^* \rangle A$.

Both classical and s-arithmetical soundness and completeness follow, of course, from theorems 3.3, and 4.4, 4.5, respectively. $\Box$

Example 5.2 Consider modality $A\mathbf{atnext}_B$ of temporal logic (cf. example 2.3). The following axioms and proof rules (together with some other ones for the nexttime operator $\bigcirc$, that can be easily derived from equations given e.g. in [10]) give sound and complete characterization of $A\mathbf{atnext}_B$.

- axioms and rules $1-3$ given in definition 3.1
- $\Gamma \Rightarrow \Sigma, G^i_{A\mathbf{atnext}_B}(\mathbf{false}), \Delta, A\mathbf{atnext}_B \vdash \Gamma \Rightarrow \Sigma, A\mathbf{atnext}_B, \Delta$,
  where i is the smallest natural number for which $G^i_{A\mathbf{atnext}_B}(\mathbf{false})$ does not appear in $\Delta^{\Gamma \Rightarrow \Sigma, A\mathbf{atnext}_B, \Delta}$
- $\{\Sigma, G^i_{A\mathbf{atnext}_B}(\mathbf{false}), \Gamma \Rightarrow \Delta\}_{i\in\omega} \vdash \Sigma, A\mathbf{atnext}_B, \Gamma \Rightarrow \Delta$.

After applying $G^i_{A\mathbf{atnext}_B}$ and making minor cosmetics, one can formulate the last two rules as follows:

- $\Gamma \Rightarrow \Sigma, \bigwedge_{0<j<i} \bigcirc^j(\neg B) \wedge \bigcirc^i(A \wedge B), \Delta, A\mathbf{atnext}_B \vdash \Gamma \Rightarrow \Sigma, A\mathbf{atnext}_B, \Delta$,
  where i is the smallest natural number for which $G^i_{A\mathbf{atnext}_B}(\mathbf{false})$ does not appear in $\Delta^{\Gamma \Rightarrow \Sigma, A\mathbf{atnext}_B, \Delta}$
- $\{\Sigma, \bigwedge_{0<j<i} \bigcirc^j(\neg B) \wedge \bigcirc^i(A \wedge B), \Gamma \Rightarrow \Delta\}_{i\in\omega} \vdash \Sigma, A\mathbf{atnext}_B, \Gamma \Rightarrow \Delta$,

The last rule, after applying rule for conjunction, can be formulated as follows:

- $\{\Sigma, \bigcirc(\neg B), ..., \bigcirc^{i-1}(\neg B), \bigcirc^i B, \bigcirc^i A, \Gamma \Rightarrow \Delta\}_{i\in\omega} \vdash \Sigma, A\mathbf{atnext}_B, \Gamma \Rightarrow \Delta$.

After replacing the last two rules by

- $\bigcirc(A \wedge B) \vee \bigcirc(\neg B \vee C) \Rightarrow C;\ \Gamma, C, \Sigma \Rightarrow \Delta \vdash \Gamma, A\mathbf{atnext}_B, \Sigma \Rightarrow \Delta$
- $C(n \leftarrow n+1) \Rightarrow \bigcirc(A \wedge B) \vee \bigcirc(\neg B \vee C(n));\ C(n \leftarrow 0) \Rightarrow \emptyset;\ \Gamma \Rightarrow \Sigma, \exists n(C(n)), \Delta \vdash \Gamma \Rightarrow \Sigma, A\mathbf{atnext}_B, \Delta$,
  where n does not appear in $A\mathbf{atnext}_B$

one obtains s-arithmetically sound and complete characterization of $A\mathbf{atnext}_B$.

Both classical and s-arithmetical soundness and completeness again follow from theorems 3.3 and 4.4, 4.5, respectively. □

6 Final remarks

As mentioned in the introduction, our axiomatizations do not deal with the greatest fixpoints. Observe, however, that one can add axioms and proof rules that, in some cases, deal with greatest fixpoints, too. Namely, assume that some $G_{w(A)}$ is downward continuous (i.e. for all $\mathcal{M}$ and v, $\mathcal{M}, v \models w(A)$ iff for all $i \in \omega$, $\mathcal{M}, v \models G^i_{w(A)}(\mathbf{true})$).

One can then add the following rules to our infinitary proof systems (cf. definition 3.1):

6. for all w defined as above,

(a) $\Gamma, G^i_{w(A)}(\mathbf{true}), \Sigma, w(A) \Rightarrow \Delta \ \vdash \ \Gamma, w(A), \Sigma \Rightarrow \Delta$,
where i is the smallest natural number for which $G^i_{w(A)}(\mathbf{true})$ does not appear in $\Gamma^{\Gamma, w(A), \Sigma \Rightarrow \Delta}$

(b) $\{\Gamma \Rightarrow \Sigma, G^i_{w(A)}(\mathbf{true}), \Delta\}_{i \in \omega} \ \vdash \ \Gamma \Rightarrow \Sigma, w(A), \Delta$.

The proofs of soundness and completeness of the obtained calculus can now be carried out as in the case of theorem 3.3.

Similarly, one can easily add suitable proof rules to proof systems defined in definition 4.1 in order to obtain s-arithmetically sound and complete proof systems:

6'. (a) $C \Rightarrow G_{w(A)}(C)$; $\Gamma \Rightarrow \Sigma, C, \Delta \ \vdash \ \Gamma \Rightarrow \Sigma, w(A), \Delta$

(b) $\emptyset \Rightarrow C(n \leftarrow 0)$; $G_{w(A)}(C(n)) \Rightarrow C(n \leftarrow n+1)$; $\Gamma, \forall n(C(n)), \Sigma \Rightarrow \Delta \vdash \Gamma, w(A), \Sigma \Rightarrow \Delta$,
where n does not appear in $w(A)$.

Observe also that the technique of infinitary proof systems we presented is applicable to the case of propositional fixpoint logics, too. In order to obtain sound and complete infinitary axiomatizations of those logics one simply has to assume axioms and rules $1, 2, 4$ and 5 of proof systems $IP_{\mathcal{L}}$ defined in definition 3.1. This also applies to propositional μ-calculus, as that has the finite model property (cf. e.g. [6]). Thus, when considering validity of μ formulas, one can restrict the class of models to finite ones only. Then all monotone functionals become both continuous and backward continuous and can thus be captured by our approach.

References

[1] H. Andréka, V. Goranko, S. Mikulas, I. Németi & I. Sain: *Effective Temporal Logics of Programs*, in: Time and Logic - A Computational Approach (L. Bolc & A. Szałas, eds.), UCL Press Ltd., 1995, 51-129.

[2] S.A. Cook: *Soundness and Completeness of Axiom System for Program Verification*, SIAM J. Comput., 7, 1, 1978, 70-90.

[3] M.J. Gordon, A.J. Milner & C.P. Wadsworth: *Edinburgh LCF*, LNCS 78, Springer-Verlag 1979.

[4] D. Harel: *Dynamic Logic*, in: Handbook of Philosophical Logic (D. Gabbay & F. Guenthner, eds.), vol. 2, D. Reidel Pub. Co., 1984, 497-607.

[5] P.G. Hinman: *Recursion Theoretic Hierarchies*, Springer-Verlag, 1978.

[6] D. Kozen: *Results on the Propositional μ-calculus*, Theoretical Computer Science, 27, 1983, 333-354.

[7] F. Kröger: *Temporal Logic of Programs*, EATCS Monographs in Comp. Sci., 8, Springer-Verlag, 1987.

[8] E.G.K. Lopez-Escobar: *An Interpolation Theorem for Denumerably Long Formulas*, Fundamenta Mathematicae, LVII, 1965, 253-272.

[9] G. Mirkowska & A. Salwicki: *Algorithmic Logic*, PWN and D. Reidel Pub. Co., 1987.

[10] A. Szałas: *On Strictly Arithmetical Completeness in Logics of Programs*, Theoretical Computer Science, 79, 1991, 341-355.

[11] A. Szałas: *Axiomatizing Fixpoint Logics*, Information Processing Letters, 41, 1992, 175-180.

[12] G. Takeuti: *Proof Theory*, North-Holland Pub. Co., 1975.