Fixed-point logic with the approximation modality and its Kripke completeness

Hiroshi Nakano

Ryukoku University, Japan
nakano@math.ryukoku.ac.jp

Abstract. We present two modal typing systems with the approximation modality, which has been proposed by the author to capture self-references involved in computer programs and their specifications. The systems are based on the simple and the F-semantics of types, respectively, and correspond to the same modal logic, which is considered the intuitionistic version of the logic of provability. We also show Kripke completeness of the modal logic and its decidability, which implies the decidability of type inhabitance in the typing systems.

1 Introduction

Although recursion, or self-reference, plays an indispensable role in both programs and their specifications, it also introduces serious difficulties into their formal treatment. Consider, for example, objects which represent integers and have an accessor method to obtain its value, and methods for doing subtraction and finding the greatest common divisor provided another integer object. In Java, the *interface*, or the coarse specification, of such objects could be written as:

```
interface Int {
    int getValue();
    Int sub(Int peer);
    Int getGCD(Int peer);
}
```

and we could implement it as the following class `Int1`, which includes some excessive occurrences of "`this`" for readability.

```
class Int1 implements Int {
    private int value;
    Int1(int v) { value = v; }        // constructor
    public int getValue() { return value; } // accessor
    public Int sub(Int peer) {        // subtraction method
      return new Int1(this.getValue() - peer.getValue());
    }
    public Int getGCD(Int peer) {     // gcd method
      if (this.getValue() == peer.getValue())
```

```
            return this;
        else if (this.getValue() > peer.getValue())
            return this.sub(peer).getGCD(peer);
        else
            return peer.getGCD(this);
    }
}
```

We could also consider another implementation, say `Int2`, which employs the following definition of `getGCD` method:

```
public Int getGCD(Int peer) {
  if (this.getValue() == peer.getValue())
      return this;
  else if (this.getValue() < peer.getValue())
      return peer.sub(this).getGCD(this);
  else
      return peer.getGCD(this);
}
```

These two class are quite symmetrical to each other, and either one works fine as long as we only use objects of the same kind. However, these two kinds of objects are not interchangeable; if we mix objects of the two classes, they run into an infinite loop whenever their `getGCD` methods are invoked with objects of the other class. If the specification being supposedly satisfied by the objects of these two classes were identical, we would be able to mix the objects without problems. So we realize that it is inevitable to give different, maybe slightly different, specifications to these two implementations of `Int` in order to obtain modularity of programs with respect to their termination, or convergence.

The approximation modality has been proposed by the author in order to incorporate general self-reference into formal specification of programs and their implementations without such loss of modularity, with which we can construct a wider range of programs, such as fixed point combinators and objects with so-called binary methods in object-oriented programming, through the proof-as-programs paradigm. We refer the reader to [1] for the motivation of the modality and examples of applications (see also [2] for proofs).

The original typing system, however, would be now considered as a specific example of a class of more general systems. In this paper, we present two basic typing systems with the modality, of which the original system can be considered an extension. One is based on the simple semantics of types, and the other is its variant based on the F-semantics of types (cf. [3, 4]). We show that both the systems have desirable convergence properties and correspond to the same modal logic, which is Kripke complete with respect to intuitionistic, transitive and converse wellfounded frames. The completeness theorem implies its decidability, and also the decidability of type inhabitance in the typing systems. We also show that the modal logic is a conservative extension of the intuitionistic version of the logic of provability (cf. [5]).

2 The typing systems

We introduce two basic modal typing systems denoted by S-$\lambda\bullet\mu$ and F-$\lambda\bullet\mu$, respectively. As a preparation for defining the syntax of type expressions, we first give one of *pseudo type expressions* **PTExp**, which are obtained by adding a unary type constructor $\bullet$ to those of $\lambda\mu$, namely the simply typed λ-calculus extended with recursive types (cf. [6, 7]). Let **TVar** be a countably infinite set of type variable symbols X, Y, Z, ... The syntax of **PTExp** is given by:

PTExp	::=	**TVar**	(type variables)
	|	**PTExp** $\to$ **PTExp**	(function types)
	|	$\bullet$ **PTExp**	(approximative types)
	|	μ**TVar**. **PTExp**	(recursive types)

Type constants are omitted for simplicity. We assume that $\to$ associates to the right as usual, and each (pseudo) type constructor associates according to the following priority: (highest) $\bullet$, $\to$, $\mu X.$ (lowest). For example, $\bullet\,\mu X.\bullet X \to Y \to Z$ is the same as $\bullet(\mu X.((\bullet X) \to (Y \to Z)))$. We use $\top$ as an abbreviation for $\mu X.\bullet X$ and use $\bullet^n A$ to denote a (pseudo) type expression $\underbrace{\bullet \ldots \bullet}_{n \text{ times}} A$, where $n \geq 0$.

Definition 1. *A type expression A is* an F-$\top$-variant *if and only if $A = \bullet^{m_0}\mu X_1.\bullet^{m_1}\mu X_2.\bullet^{m_2}\ldots\mu X_n.\bullet^{m_n} X_i$ for some n, m_0, m_1, m_2, ..., m_n, X_1, X_2, ..., X_n and i such that $1 \leq i \leq n$ and $m_i + m_{i+1} + m_{i+2} + \ldots + m_n \geq 1$. A type expression A is* an S-$\top$-variant *if and only if $\overline{A}$ is an F-$\top$-variant, where $\overline{A}$ is defined as follows:*

$$\overline{X} = X, \quad \overline{A \to B} = \overline{B}, \quad \overline{\mu X.A} = \mu X.\overline{A}.$$

An F-$\top$-variant is also an S-$\top$-variant, and by definition it is decidable whether a type expression is an S(F)-$\top$-variant or not. S(F)-$\top$-variants correspond to the universe into which λ-terms are interpretated. Hence, every λ-term should have these types in S(F)-$\lambda\bullet\mu$, respectively.

Definition 2. *We say that a pseudo type expression A is* S-proper *(respectively* F-proper*) in X if and only if X occurs freely only (a) in scopes of the $\bullet$-operator in A, or (b) in a subexpression $B \to C$ of A with C being an S-$\top$-variant (F-$\top$-variant).*[1]

For example, $\bullet X$, $\bullet(X \to Y)$, $\mu Y.\bullet(X \to Y)$, and $X \to \top$ are S(F)-proper in X, and neither X, $X \to Y$ nor $\mu Y.\mu Z.X \to Y$ is S(F)-proper in X.

Definition 3. *A* type expression *of S-$\lambda\bullet\mu$ (respectively F-$\lambda\bullet\mu$) is a pseudo type expression such that A is S-proper (F-proper) in X for any of its subexpressions in the form of $\mu X.A$. We denote the set of type expressions by* **TExp**.

[1] The condition (b) is included so that the equivalence relation $\simeq$ on type expressions (cf. Definition 4) preserves properness.

For example, X, $X \to Y$, $\mu X.\bullet X \to Y$, $\mu X.X \to \top$ and $\mu X.\bullet\, \mu Y.X \to Z$ are type expressions, and neither $\mu X.X \to Y$ nor $\mu X.\mu Y.X \to Y$ is a type expression. We use $A, B, C, D, \ldots$ to denote type expressions of $\lambda\bullet\mu$'s, and denote the set of type variables occurring freely in A by $FTV(A)$ regarding a type variable X as bound in $\mu X.A$. We also regard α-convertible type expressions as identical, and use $A[B_1/X_1, \ldots, B_n/X_n]$ to denote the type expression obtained from A by substituting $B_1, \ldots, B_n$ for each free occurrence of $X_1, \ldots, X_n$, respectively.

Definition 4. *The equivalence relation $\simeq$ on type expressions is defined as the smallest binary relation that satisfies:*

($\simeq$-*reflex*) $A \simeq A$.
($\simeq$-*symm*) *If* $A \simeq B$, *then* $B \simeq A$.
($\simeq$-*trans*) *If* $A \simeq B$ *and* $B \simeq C$, *then* $A \simeq C$.
($\simeq$-$\bullet$) *If* $A \simeq B$, *then* $\bullet A \simeq \bullet B$.
($\simeq$-$\to$) *If* $A \simeq C$ *and* $B \simeq D$, *then* $A \to B \simeq C \to D$.
($\simeq$-*fix*) $\mu X.A \simeq A[\mu X.A/X]$.
($\simeq$-*uniq*) *If* $A \simeq C[A/X]$ *and* C *is S(F)-proper in* X, *then* $A \simeq \mu X.C$.

All the condition above are common to S-$\lambda\bullet\mu$ and F-$\lambda\bullet\mu$, and the following ones are respectively satisfied:

S-$\lambda\bullet\mu$: ($\simeq$-$\to\top$) $A \to \top \simeq \top$.
F-$\lambda\bullet\mu$: ($\simeq$-$\to\top$) $A \to \top \simeq B \to \top$.

Intuitively, two type expressions are equivalent if their (possibly infinite) type expression obtained by unfolding recursive types occurring in them indefinitely are identical modulo the rule ($\simeq$-$\to\top$). This equality on type expressions is decidable. One can also observe that a type expression A is an S(F)-$\top$-variant if and only if $A \simeq \top$ in S(F)-$\lambda\bullet\mu$, respectively.

We now define a subtyping relation on type expressions, which is induced by the $\bullet$-modality, by a set of the subtyping rules (cf. [8]). A *subtyping assumption* is a finite set of pairs of type variables such that any type variable appears at most once in the set. We write $\{X_1 \preceq Y_1, X_2 \preceq Y_2, \ldots, X_n \preceq Y_n\}$ to denote the subtyping assumption $\{ <X_i, Y_i> \mid i = 1, 2, \ldots, n \}$, and use γ, γ', γ_1, γ_2, ... to denote subtyping assumptions, and $FTV(\gamma)$ to denote the set of type variables occurring in γ.

Definition 5 ($\preceq$). *The derivability of* a subtyping judgment $\gamma \vdash A \preceq B$ *is defined by the following subtyping rules:*

$$\frac{}{\gamma \cup \{X \preceq Y\} \vdash X \preceq Y}\ (\preceq\text{-}assump) \qquad \frac{}{\gamma \vdash A \preceq \top}\ (\preceq\text{-}\top)$$

$$\frac{}{\gamma \vdash A \preceq \bullet A}\ (\preceq\text{-}approx) \qquad \frac{}{\gamma \vdash A \preceq A'}\ (\preceq\text{-}reflex)\quad (A \simeq A')$$

$$\frac{\gamma_1 \vdash A \preceq B \quad \gamma_2 \vdash B \preceq C}{\gamma_1 \cup \gamma_2 \vdash A \preceq C}\ (\preceq\text{-}trans)$$

$$\frac{\gamma \vdash A \preceq B}{\gamma \vdash \bullet A \preceq \bullet B}\ (\preceq\text{-}\bullet) \qquad \frac{\gamma_1 \vdash A' \preceq A \quad \gamma_2 \vdash B \preceq B'}{\gamma_1 \cup \gamma_2 \vdash A \to B \preceq A' \to B'}\ (\preceq\text{-}\to)$$

$$\frac{\gamma \cup \{X \preceq Y\} \vdash A \preceq B}{\gamma \vdash \mu X.A \preceq \mu Y.B}\ (\preceq\text{-}\mu) \quad \left(\begin{array}{l} X \notin FTV(\gamma) \cup FTV(B),\ Y \notin FTV(\gamma) \cup \\ FTV(A),\ \textit{and}\ A\ \textit{and}\ B\ \textit{are S(F)-proper} \\ \textit{in}\ X\ \textit{and}\ Y,\ \textit{respectively} \end{array}\right)$$

Note that $\gamma \cup \{X \preceq Y\}$ and $\gamma_1 \cup \gamma_2$ in the rules above must be (valid) subtyping assumptions, i.e., any type variable must not have more than one occurrence in them. All the rules above are common to S-$\lambda\bullet\mu$ and F-$\lambda\bullet\mu$, and they respectively have another rule called $(\to\preceq\text{-}\to\bullet)$ as follows:

$$\text{S-}\lambda\bullet\mu: \frac{}{\gamma \vdash \bullet(A \to B) \preceq \bullet A \to \bullet B}\ (\preceq\text{-}\to\bullet)$$

$$\text{F-}\lambda\bullet\mu: \frac{}{\gamma \vdash A \to B \preceq \bullet A \to \bullet B}\ (\preceq\text{-}\to\bullet)$$

The binary relation $\preceq$ on type expressions is defined as: $A \preceq B$ if and only if $\{\} \vdash A \preceq B$ is derivable. It should be noted that if $A \preceq B$ in F-$\lambda\bullet\mu$, then it is also the case in S-$\lambda\bullet\mu$.

We now define the typing rules for S-$\lambda\bullet\mu$ and F-$\lambda\bullet\mu$. A *typing context* is a finite mapping that assigns a type expression to each individual variable of its domain. We use $\Gamma, \Gamma', \ldots$ to denote typing contexts, and $\{x_1 : A_1, \ldots, x_m : A_m\}$ to denote the typing context that assigns A_i to x_i for every i. We write $\Gamma' \preceq \Gamma$ if and only if $Dom(\Gamma'(x)) = Dom(\Gamma(x))$ and $\Gamma'(x) \preceq \Gamma(x)$ for every $x \in Dom(\Gamma)$.

Definition 6. *The typing systems S-$\lambda\bullet\mu$ and F-$\lambda\bullet\mu$ are defined by the following derivation rules:*

$$\frac{}{\Gamma \cup \{x : A\} \vdash x : A}\ (var) \qquad \frac{\Gamma \vdash M : A}{\bullet\Gamma \vdash M : \bullet A}\ (nec)$$

$$\frac{}{\Gamma \vdash M : \top}\ (\top) \qquad \frac{\Gamma \vdash M : A \quad \Gamma'(x) \preceq \Gamma(x) \quad A \preceq A'}{\Gamma' \vdash M : A'}\ (\preceq)$$

$$\frac{\Gamma \cup \{x : A\} \vdash M : B}{\Gamma \vdash \lambda x.\, M : A \to B}\ (\to I) \qquad \frac{\Gamma_1 \vdash M : \bullet^n(A \to B) \quad \Gamma_2 \vdash N : \bullet^n A}{\Gamma_1 \cup \Gamma_2 \vdash MN : \bullet^n B}\ (\to E)$$

where $Dom(\bullet\Gamma) = Dom(\Gamma)$ and $(\bullet\Gamma)(x) = \bullet\Gamma(x)$ for every $x \in Dom(\Gamma)$. Note that since S-$\lambda\bullet\mu$ has the subtyping rule $\bullet(A \to B) \preceq \bullet A \to \bullet B$, the ($\to$E)-rule for S-$\lambda\bullet\mu$ can be simplified to the following usual form:

$$\frac{\Gamma_1 \vdash M : A \to B \quad \Gamma_2 \vdash N : A}{\Gamma_1 \cup \Gamma_2 \vdash MN : B}\ (\to\text{E})$$

Since $A \preceq B$ in F-$\lambda\bullet\mu$ implies the same in S-$\lambda\bullet\mu$, one can observe the following.

Proposition 1. *If* $\Gamma \vdash M : A$ *is derivable in F-*$\lambda\bullet\mu$*, then so is it in S-*$\lambda\bullet\mu$*.*

The most interesting thing about S(F)-$\lambda\bullet\mu$ is that one can derive $\vdash \mathbf{Y} : (\bullet A \to A) \to A$ for any A, where $\mathbf{Y} = \lambda f.\,(\lambda x.\,f(xx))\,(\lambda x.\,f(xx))$ (cf. [1]). The typing systems S-$\lambda\bullet\mu$ and F-$\lambda\bullet\mu$ also enjoy some basic properties such as subject reduction property.

Proposition 2. *(1) If* $\Gamma \vdash M : A$ *is derivable, then* $FV(M) \subset Dom(\Gamma)$*.*

(2) If $\Gamma \cup \{x : A\} \vdash M : B$ *and* $\Gamma \vdash N : A$ *are derivable, then so is* $\Gamma \vdash M[N/x] : B$*.*

(3) If $\Gamma \vdash M : A$ *is derivable and* $M \underset{\beta}{\to} M'$*, then* $\Gamma \vdash M' : A$ *is derivable.*

Proof. Straightforward induction on the derivations. In the proof of (3), we apply the following property of $\preceq$ to the case that the derivation ends with ($\to$E): if $A \to B \preceq \bullet^n(C \to D)$ and $D \not\preceq \top$, then $\bullet^l B \preceq C$ and $D \preceq \bullet^l A$ for some l. □

3 Semantics

In this section, we show revised results presented in Sections 4 and 5 of [1]. We give two kinds of realizability interpretations, the simple semantics and the F-semantics, over certain Kripke-frames to S-$\lambda\bullet\mu$ and F-$\lambda\bullet\mu$, respectively, and show soundness of each typing system with respect to the corresponding interpretation. We also show that the new systems preserve the convergence properties of well-typed λ-terms presented in [1].

We now consider the following class of Kripke-frames.

Definition 7. *A transitive and converse wellfounded frame is a pair* $<\mathcal{W}, \to>$*, which consists of a set* $\mathcal{W}$ *of* possible worlds *and* an accessibility relation $\to$ *on* $\mathcal{W}$ *such that:*

(1) The relation $\to$ *is transitive.*

(2) The relation $\to$ *is converse wellfounded, i.e., there is no infinite sequence such that* $p_0 \to p_1 \to p_2 \to p_3 \to \ldots$*.*

Let $<\mathcal{V}, \cdot, [\![\ ]\!]>$ be a λ-model of untyped λ-calculus. The meaning of a λ-term M is denoted by $[\![M]\!]_\rho$, where ρ is *an individual environment* that assigns an element of $\mathcal{V}$ to each individual variable. Each type expression is interpreted as a mapping $\mathcal{I}$ from $\mathcal{W}$ to the power set $\mathcal{P}(\mathcal{V})$ of $\mathcal{V}$ such that:

$$p \to q \quad \text{implies} \quad \mathcal{I}(p) \subset \mathcal{I}(q)$$

A mapping that assigns such a monotone mapping to each type variable is called *a type environment.*

Definition 8 (Semantics of types). *Let* $<\mathcal{W}, \to>$ *be a transitive and converse wellfounded frame, and* ξ *a type environment. We define a mapping* $\mathcal{I}_S(A)^\xi$

from $\mathcal{W}$ to $\mathcal{P}(\mathcal{V})$ for each type expression A by extending ξ as follows, where we prefer to write $\mathcal{I}_S(A)^\xi_p$ rather than $\mathcal{I}_S(A)^\xi(p)$.

$$\begin{aligned}
\mathcal{I}_S(X)^\xi_p &= \xi(X)_p \\
\mathcal{I}_S(\bullet A)^\xi_p &= \{\ u \mid u \in \mathcal{I}_S(A)^\xi_q \ \ \textit{for every } q \leftarrow p\ \} \\
\mathcal{I}_S(A \to B)^\xi_p &= \left\{ u \;\middle|\; \begin{array}{l} \textit{If } B \textit{ is not an S-}\top\textit{-variant, then } u \cdot v \in \mathcal{I}_S(B)^\xi_q \\ \textit{for every } v \in \mathcal{I}_S(A)^\xi_q \textit{ whenever } q = p \textit{ or } q \leftarrow p. \end{array} \right\} \\
\mathcal{I}_S(\mu X.A)^\xi_p &= \mathcal{I}_S(A[\mu X.A/X])^\xi_p
\end{aligned}$$

$\mathcal{I}_S$ is called *the simple semantics* of types. We similarly define $\mathcal{I}_F(A)^\xi$, *the F-semantics* of types, where the only difference is the definition of $\mathcal{I}_F(A \to B)^\xi$, which is defined as:

$$\mathcal{I}_F(A \to B)^\xi_p = \left\{ u \;\middle|\; \begin{array}{l} \text{1. If } B \text{ is not an F-}\top\text{-variant, then } u \cdot v \in \mathcal{I}_F(B)^\xi_q \\ \quad \text{for every } v \in \mathcal{I}_F(A)^\xi_q \text{ whenever } q = p \text{ or } q \leftarrow p, \\ \quad \text{and} \\ \text{2. } u = [\![\lambda x.\, M]\!]_\rho \text{ for some } x, \rho \text{ and } M. \end{array} \right\}$$

In the squeal, we prefer to write $\mathcal{I}(A)^\xi$, or $\mathcal{I}(A)$, rather than $\mathcal{I}_S(A)^\xi$ or $\mathcal{I}_F(A)^\xi$ when it would cause no confusion in context. Note that the $\mathcal{I}(A)^\xi_p$ has been defined by induction on the lexicographic ordering of $<p, r(A)>$, where the non-negative integer $r(A)$ is defined as:

$$\begin{aligned}
&r(X) = r(\bullet A) = 0 \\
&r(A \to B) = \begin{cases} 0 & (B \text{ is an S(F)-}\top\text{-variant}) \\ \max(r(A), r(B)) + 1 & (\text{otherwise}) \end{cases} \\
&r(\mu X.A) = r(A) + 1
\end{aligned}$$

$\mathcal{I}(\mu X.A)^\xi_p$ is well defined since $r(A[B/X]) < r(\mu X.A)$ for any B whenever A is S(F)-proper in X. We can easily verify that $p \to q$ implies $\mathcal{I}(A)^\xi_p \subset \mathcal{I}(A)^\xi_q$.

Proposition 3. *The equivalence relation $\simeq$ and the subtyping relation $\preceq$ on type expressions well respect these semantics. That is:*

(1) If $A \simeq B$, then $\mathcal{I}(A)^\xi_p = \mathcal{I}(B)^\xi_p$ for every $p \in \mathcal{W}$.
(2) If $A \preceq B$, then $\mathcal{I}(A)^\xi_p \subset \mathcal{I}(B)^\xi_p$ for every $p \in \mathcal{W}$.

From these results, we can also show the soundness of S-$\lambda\bullet\mu$ and F-$\lambda\bullet\mu$ with respect to the semantics of types $\mathcal{I}_S$ and $\mathcal{I}_F$, respectively.

Theorem 1 (Soundness). *If $\{x_1 : A_1, \ldots, x_n : A_n\} \vdash M : B$ is derivable, then $[\![M]\!]_\rho \in \mathcal{I}(B)^\xi_p$ for every p, ξ and ρ whenever $\rho(x_i) \in \mathcal{I}(A_i)^\xi_p$ for every i ($i = 1, 2, \ldots, n$).*

Proof. By induction on the derivation and by cases of the last rule used in the derivation. Most cases are straightforward. Use Proposition 3 for the case of ($\preceq$). Prove it by induction on p in the case of ($\to$I). □

One can observe that F-$\lambda\bullet\mu$ is also sound with respect to the simple semantics $\mathcal{I}_S$ by Proposition 1. If the transitive and converse wellfounded frame $<\mathcal{W}, \rightarrow>$ also satisfies the following extra condition:

> if $r \rightarrow p$, then $r \xrightarrow{*} q \rightarrow p$ for some q such that $q \rightarrow s$ implies $p \xrightarrow{*} s$ for any s, where $\xrightarrow{*}$ denotes the reflexive (and transitive) closure of $\rightarrow$,

then the rule below is also sound with respect to $\mathcal{I}_S$ (respectively $\mathcal{I}_F$), when added to S-$\lambda\bullet\mu$ (F-$\lambda\bullet\mu$).

$$\frac{}{\Gamma \vdash \bullet A \rightarrow \bullet B \preceq \bullet(A \rightarrow B)} \ (\preceq\text{-}\bullet\rightarrow)$$

Similarly, if for every $p \in \mathcal{W}$ there exists some $q \in \mathcal{W}$ such that $q \rightarrow p$, then

$$\frac{\bullet\Gamma \vdash M : \bullet A}{\Gamma \vdash M : A} \ (\bullet)$$

is sound. For example, the set of non-negative integers, or limit ordinals, and the "greater than" relation $>$, where a smaller number is accessible from a larger one, constitute a frame satisfying the two conditions above. We call the extended systems with these two rules S-$\lambda\bullet\mu^+$ and F-$\lambda\bullet\mu^+$, respectively, where the $(\preceq\text{-}\bullet\rightarrow)$ rule makes (nec) redundant. It should be noted that the two rules provide the converses of $(\preceq\text{-}\rightarrow\bullet)$ and (nec), respectively. The original system given in [1] is equivalent to F-$\lambda\bullet\mu^+$. Although the base systems S-$\lambda\bullet\mu$ and F-$\lambda\bullet\mu$ are somewhat weaker than that, all the examples of programs presented in the paper still work in them.

Theorem 1 assures us that the modularity of programs is preserved even if we regard type expressions, or specifications, as asserting the convergence of programs. For example, if a type B comprises of certain canonical values, and we have a program M of a type $A \rightarrow B$, then we can expect that M terminates and returns such a canonical value when we provide a value of A. By a discussion on soundness with respect to an interpretation over the term model of untyped λ-calculus, we can obtain such convergence properties of well-typed λ-terms. The corresponding results for the original system F-$\lambda\bullet\mu^+$ was first presented in Section 5 of [1].

Definition 9. *A type expression A is* tail finite *if and only if $A \simeq \bullet^{m_1}(B_1 \rightarrow \bullet^{m_2}(B_2 \rightarrow \bullet^{m_3}(B_3 \rightarrow \ldots \rightarrow \bullet^{m_n}(B_n \rightarrow X)\ldots)))$ for some n, m_0, m_1, m_2, …, m_n, B_1, B_2, …, B_n and X.*

A type expression is tail finite if and only if it is not an S-$\top$-variant.

Definition 10. *Let A be a type expression. Two sets $ETV^+(A)$ and $ETV^-(A)$ of type variables are defined as follows:*

$$ETV^+(X) = \{X\}, \quad ETV^-(X) = \{\},$$
$$ETV^{\pm}(\bullet A) = ETV^{\pm}(A),$$
$$ETV^{\pm}(A \to B) = \begin{cases} \{\} & \textit{(B is an S(F)-$\top$-variant)} \\ ETV^{\mp}(A) \cup ETV^{\pm}(B) & \textit{(otherwise)} \end{cases}$$
$$ETV^{\pm}(\mu X.A) = \begin{cases} (ETV^{\pm}(A) \cup ETV^{\mp}(A)) - \{X\} & \textit{($X \in ETV^-(A)$)} \\ ETV^{\pm}(A) - \{X\} & \textit{(otherwise)} \end{cases}$$

It should be noted that the set $ETV^+(A)$ ($ETV^-(A)$) consists of the type variables that have free positive (negative) occurrences in A, where we ignore any subexpression $B \to C$ of A whenever C is an S(F)-$\top$-variant. If $X \in ETV^{\pm}(A)$ in S-$\lambda\bullet\mu$, then so is in F-$\lambda\bullet\mu$.

Definition 11. *A type expression A is* positively (negatively) finite *if and only if C is tail finite whenever $A \simeq B[C/X]$ for some B and X such that $X \in ETV^+(B)$ ($X \in ETV^-(B)$) and $X \notin ETV^-(B)$ ($X \notin ETV^+(B)$).*

Every positively finite type expression is tail finite. If a type expression of F-$\lambda\bullet\mu$ is tail (positively, or negatively) finite, then so is as a type expression of S-$\lambda\bullet\mu$.

Theorem 2 (Convergence). *Let $\Gamma \vdash M : A$ be derivable in S-$\lambda\bullet\mu$, F-$\lambda\bullet\mu$, S-$\lambda\bullet\mu^+$ or F-$\lambda\bullet\mu^+$.*

(1) If A is tail finite, then M is head normalizable.
(2) If A is positively finite, and $\Gamma(x)$ is negatively finite for every $x \in Dom(\Gamma)$, then the Böhm tree of M has no occurrence of $\bot$, i.e., a λ-term not being head normalizable.

Proof. It suffices to prove the case of S-$\lambda\bullet\mu^+$. See Appendix. □

Moreover, if the typing judgement is derivable in F-$\lambda\bullet\mu$ or F-$\lambda\bullet\mu^+$ for some A not being an F-$\top$-variant, then M is weakly head normalizable, i.e., β-reduces to the form $\lambda x.\, N$ or $x\, N_1\, N_2\, \ldots\, N_n$ $(n \geq 0)$ (cf. [1]).

4 The modal logic behind S-$\lambda\bullet\mu$ and F-$\lambda\bullet\mu$

In this section, we consider S-$\lambda\bullet\mu$ and F-$\lambda\bullet\mu$ as modal logics by ignoring left hand sides of ":" from typing judgments, and show that they precisely correspond to the same modal logic.

Definition 12 (Formal system $L\bullet\mu$). *We define a modal logic considering type expressions as logical formulae, where the equivalence relation $\simeq_L$ on formulae is defined as the smallest binary relation that satisfies the conditions listed*

in Definition 4 except ($\simeq$*-uniq*) *and* ($\simeq$-$\rightarrow\top$). *Let* $L\bullet\mu$ *be the formal system defined by the following inference rules, where* Γ *denotes a finite set of formulae.*

$$\frac{}{\Gamma \cup \{A\} \vdash A}\ (assump) \qquad \frac{\Gamma \vdash A}{\bullet\Gamma \vdash \bullet A}\ (nec) \qquad \frac{\Gamma \vdash A}{\Gamma \vdash A'}\ (\simeq_L)\quad (A \simeq_L A')$$

$$\frac{}{\Gamma \vdash \bullet(A \to B) \to \bullet A \to \bullet B}\ (K) \qquad \frac{}{\Gamma \vdash A \to \bullet A}\ (approx)$$

$$\frac{\Gamma \cup \{A\} \vdash B}{\Gamma \vdash A \to B}\ (\to I) \qquad \frac{\Gamma_1 \vdash A \to B \quad \Gamma_2 \vdash A}{\Gamma_1 \cup \Gamma_2 \vdash B}\ (\to E)$$

Proposition 4. *If* $\{A_1, \ldots, A_n\} \vdash B$ *is derivable in* $L\bullet\mu$, *then* $\{x_1 : A_1, \ldots, x_n : A_n\} \vdash M : B$ *is derivable in F-*$\lambda\bullet\mu$ *for some* λ*-term* M *and distinct individual variables* $x_1, \ldots, x_n$, *such that* $FV(M) \subset \{x_1, \ldots, x_n\}$.

Proof. Straightforward. □

Definition 13. *A* $\bullet\mu$*-frame is a triple* $<\mathcal{W}, \rightarrow, R>$, *which consists of a set* $\mathcal{W}$ *of possible worlds and two accessibility relations* $\rightarrow$ *and* R *on* $\mathcal{W}$ *such that:*

(1) $<\mathcal{W}, R>$ *is a transitive and converse wellfounded frame.*
(2) $\rightarrow$ *is a transitive relation on* $\mathcal{W}$.
(3) $p \rightarrow q$ *implies* $p\ R\ q$.

It should be noted that $\rightarrow$ is also converse wellfounded by the condition (3); and hence $<\mathcal{W}, \rightarrow>$ is also a transitive and converse wellfounded frame.

Definition 14 (Semantics of $L\bullet\mu$**).** *Let* $<\mathcal{W}, \rightarrow, R>$ *be a* $\bullet\mu$*-frame. A mapping* $\mathcal{I}$ *from* $\mathcal{W}$ *to* $\{\mathbf{t}, \mathbf{f}\}$ *is* hereditary *if and only if:*

$$\text{if } p\ R\ q, \text{ then } \mathcal{I}(p) = \mathbf{t} \text{ implies } \mathcal{I}(q) = \mathbf{t}.$$

A mapping ξ *that assigns a hereditary mapping to each propositional variable, i.e., type variable, is called* a valuation. *We define a hereditary mapping* $\mathcal{I}_L(A)^\xi$ *from* $\mathcal{W}$ *to* $\{\mathbf{t}, \mathbf{f}\}$ *for each formula* A *by extending* ξ *as follows, where we write* $\models^\xi_p A$ *to denote* $\mathcal{I}_L(A)^\xi(p) = \mathbf{t}$.

$\models^\xi_p X$ *iff* $\xi(X)_p = \mathbf{t}$
$\models^\xi_p A \to B$ *iff* $\models^\xi_q A$ *implies* $\models^\xi_q B$ *for every* q *such that* $q = p$ *or* $p\ R\ q$
$\models^\xi_p \bullet A$ *iff* $\models^\xi_q A$ *for every* q *such that* $p \rightarrow q$ *or* $p\ R\ r \rightarrow q$ *for some* r
$\models^\xi_p \mu X.A$ *iff* $\models^\xi_p A[\mu X.A/X]$

Note that $\models^\xi_p A$ is again defined by induction on the lexicographic ordering of $<p, r(A)>$. We write $\Gamma \models^\xi_p A$ if and only if $\models^\xi_p A$ whenever $\models^\xi_p B$ for every $B \in \Gamma$. By a discussion similar to Theorem 1, one observes soundness of S-$\lambda\bullet\mu$ as a logic with respect to this semantics of formulae.

Proposition 5. *If* $A \preceq B$ *in S-*$\lambda\bullet\mu$*, then* $\{A\} \models_p^\xi B$.

Proposition 6. *Let* $<\mathcal{W}, \rightarrow, R>$ *be a* $\bullet\mu$*-frame, and* ξ *a valuation. If* $\{x_1 : A_1, \ldots, x_n : A_n\} \vdash M : B$ *is derivable in S-*$\lambda\bullet\mu$*, then* $\{A_1, \ldots, A_n\} \models_p^\xi B$ *for every* $p \in \mathcal{W}$.

The main results of the present paper can be summarized as the following theorem.

Theorem 3. *The following four conditions are equivalent.*

(1) $\{A_1, , \ldots, A_n\} \vdash B$ *is derivable in* $L\bullet\mu$.
(2) $\{x_1 : A_1, \ldots, x_n : A_n\} \vdash M : B$ *is derivable in F-*$\lambda\bullet\mu$ *for some* M, x_1, $\ldots$, x_n.
(3) $\{x_1 : A_1, \ldots, x_n : A_n\} \vdash M : B$ *is derivable in S-*$\lambda\bullet\mu$ *for some* M, x_1, $\ldots$, x_n.
(4) $\{A_1, , \ldots, A_n\} \models_p^\xi B$ *for every* $\bullet\mu$*-frame* $<\mathcal{W}, \rightarrow, R>$, *valuation* ξ, *and* $p \in \mathcal{W}$.

Proof. We get (1) $\Rightarrow$ (2), (2) $\Rightarrow$ (3), and (3) $\Rightarrow$ (4) by Propositions 4, 1 and 6, respectively. Hence, it suffices to show that (4) $\Rightarrow$ (1), which is given by the following completeness theorem. □

Theorem 4 (Completeness of $L\bullet\mu$**).** *If* $\{A_1, \ldots, A_n\} \vdash M : B$ *is not derivable in* $L\bullet\mu$*, then there exist some* $\bullet\mu$*-frame* $<\mathcal{W}_0, \rightarrow_0, R_0>$, *valuation* ξ_0, *and* $p_0 \in \mathcal{W}_0$ *such that* $\not\models_{p_0}^{\xi_0} B$ *while* $\models_{p_0}^{\xi_0} A_i$ *for every* i $(i = 1, 2, \ldots, n)$.

The rest of the present section is devoted to proving this theorem. Suppose that $\{A_1, \ldots, A_n\} \vdash M : B$ is not derivable.

Let C and D be formulae, i.e., type expressions. We call C a *component* of D, and write $C \leq D$, if and only if

$$E[C/X] \simeq_L D \quad \text{and} \quad X \in FTV(E)$$

for some type expression E and type variable X. We also define $Comp(D)$ as:

$$Comp(D) = \{C \mid C \leq D\}.$$

Note that $Comp(D)/\simeq_L$ is a finite set (cf. e.g. [9, 8]). Let

$$\mathcal{F} = \{\ C \mid C \in Comp(B) \text{ or } C \in Comp(A_i) \text{ for some } i\ \},$$

and define W_0 and p_0 as:

$$W_0 = \{\ p \subset \mathcal{F} \mid C \in p \text{ whenever } C \in \mathcal{F} \text{ and } p \vdash C \text{ is derivable}^2\ \}$$
$$p_0 = \{\ C \in \mathcal{F} \mid \{A_1, \ldots, A_n\} \vdash C \text{ is derivable}\ \}$$

Note that $\mathcal{W}_0$ is a finite set since $Comp(D)/\simeq_L$ is finite and $L\bullet\mu$ has the $(\simeq_L)$ rule. Then, for each $p \in W_0$, define $\tilde{p}$ as:

$$\tilde{p} = \{\ C \in \mathcal{F} \mid p \vdash \bullet C \text{ is derivable}\ \}$$

[2] More precisely, $\Gamma' \vdash C$ derivable for some finite $\Gamma' \subset p$.

Observe that $p \in \mathcal{W}_0$ implies $\tilde{p} \in \mathcal{W}_0$, since if $\tilde{p} \vdash C$ is derivable for some $C \in \mathcal{F}$, then so is $\bullet\tilde{p} \vdash \bullet C$ by (nec); and therefore, $p \vdash \bullet C$ is also derivable, i.e., $C \in \tilde{p}$. Note also that $p \subset \tilde{p}$ holds because $L\bullet\mu$ has the (approx) rule. The accessibility relations $\rightarrow_0$ and R_0 are defined as follows:

$$\begin{aligned} p \rightarrow_0 q \quad &\text{iff} \quad \tilde{p} \subset q \text{ and } \tilde{q} \neq q. \\ p \; R_0 \; q \quad &\text{iff} \quad p \subset q \text{ and } p \neq q. \end{aligned}$$

We can easily verify that $\rightarrow_0$ and R_0 are transitive, and $p \rightarrow_0 q$ implies $p \; R_0 \; q$. Since $\mathcal{W}_0$ is finite, R_0 is also converse wellfounded. We finally define the valuation ξ_0 as:

$$\xi_0(X)_p = \begin{cases} \mathbf{t} & (X \in p) \\ \mathbf{f} & (X \notin p) \end{cases}$$

Obviously, ξ_0 is hereditary by the definition of R_0. Since $B \notin p_0$ while $A_i \in p_0$ for every i, to finish the proof of the completeness theorem, it suffices to prove the following lemma.

Lemma 1. *Let $C \in \mathcal{F}$ and $p \in \mathcal{W}_0$. Then, $C \in p$ if and only if $\models^{\xi_0}_p C$.*

Proof. The proof proceeds by induction on the lexicographic ordering of $<p, r(C)>$, and by cases of the form of C.

Case: $C = X$. Trivial from the definition of $\xi_0(X)$.

Case: $C = D \rightarrow E$. For the "only if" part, suppose that $D \rightarrow E \in p$, $\models^{\xi_0}_q D$, and $q = p$ or $p \; R_0 \; q$. We get $D \in q$ from $\models^{\xi_0}_q D$ by induction hypothesis, since p decreases to q or else $r(D) < r(D \rightarrow E)$, and $D \rightarrow E \in q$ from $p \subset q$. Therefore, $E \in q$, and by induction hypothesis again, $\models^{\xi_0}_q E$. Thus we get $\models^{\xi_0}_p D \rightarrow E$. As for "if" part, suppose that $\models^{\xi_0}_p D \rightarrow E$, i.e.,

$$\models^{\xi_0}_q D \;\text{ implies }\; \models^{\xi_0}_q E \;\text{ whenever }\; q = p \;\text{ or }\; p \; R_0 \; q \tag{1}$$

Let q as:

$$q = \{\, C' \in \mathcal{F} \mid p \cup \{D\} \vdash C' \text{ is derivable} \,\}.$$

Note that $q = p$ or $p \; R_0 \; q$. Since $D \in q$, we get $\models^{\xi_0}_q D$ by induction hypothesis, and then $\models^{\xi_0}_q E$ from (1). Hence, by induction hypothesis again, $E \in q$, i.e., $p \cup \{D\} \vdash E$ is derivable, and so is $p \vdash D \rightarrow E$.

Case: $C = \bullet D$. For the "only if" part, suppose that $\bullet D \in p$. If $p \rightarrow q$ or $p \; R \; r \rightarrow q$ for some r, then since $\tilde{p} \subset q$ and $p \vdash \bullet D$ is derivable, we get $D \in q$. Hence, $\models^{\xi_0}_q D$ by induction hypothesis. We thus get $\models^{\xi_0}_p \bullet D$. For "if" part, suppose that $\models^{\xi_0}_p \bullet D$, i.e.,

$$\models^{\xi_0}_q D \;\text{ for any } q \;\text{ if } p \rightarrow q \text{ or } p \; R \; r \rightarrow q \text{ for some } r\,. \tag{2}$$

Let q as:

$$q = \{\ C' \in \mathcal{F} \mid \tilde{p} \cup \{\bullet D\} \vdash C' \text{ is derivable }\}.$$

If $p \to q$, then $\models_q^{\xi_0} D$ by (2); therefore, $D \in q$ by induction hypothesis. Otherwise, $\tilde{q} = q$, i.e., also $D \in q$. Hence, $\tilde{p} \cup \{\bullet D\} \vdash D$ is derivable. On the other hand, there is a derivation of $\vdash (\bullet D \to D) \to D$ corresponding to the **Y**-combinator. Therefore, $\tilde{p} \vdash D$ is also derivable, and so is $\bullet\tilde{p} \vdash \bullet D$ by (nec). That is, $p \vdash \bullet D$ is derivable; and therefore, $\bullet D \in p$.

Case: $C = \mu X.D$. For the "only if" part, suppose that $\mu X.D \in p$, i.e., also $D[\mu X.D/X] \in p$ by ($\simeq_L$) rule. We get $\models_p^{\xi_0} D[\mu X.D/X]$ by induction hypothesis, since $r(D[\mu X.D/X]) < r(\mu X.D)$; and therefore, $\models_p^{\xi_0} \mu X.D$ by definition. For "if" part, suppose that $\models_p^{\xi_0} \mu X.D$, i.e., $\models_p^{\xi_0} D[\mu X.D/X]$. We get $D[\mu X.D/X] \in p$ by induction hypothesis; and therefore, $\mu X.D \in p$ by the ($\simeq_L$) rule. □

This completes the proof of Theorems 4 and 3. Since the counter model constructed in the proof of Lemma 1 is based on a finite frame, the logic $L\bullet\mu$ has the finite model property, and we therefore get the following corollary.

Corollary 1. *The following problems are decidable.*

(1) Provability in $L\bullet\mu$.
(2) Type inhabitance in S-$\lambda\bullet\mu$.
(3) Type inhabitance in F-$\lambda\bullet\mu$.

5 Relationship to the intuitionistic logic of provability

The logic $L\bullet\mu$ permits self-referential formulae. In this section, we show that if $L\bullet\mu$ is restricted to finite formulae, i.e., those without any occurrence of μ, then one gets the *intuitionistic* version of the logic of provability GL (cf. [5]), where "intuitionistic" means that the interpretation is monotonic with respect to the accessibility relation, and not provability in intuitionistic systems such as HA. GL is also denoted by G (for Gödel), L (for Löb), PrL, KW, or K4W, in the literature.

Definition 15 (Formal system iKW). *We define a modal logic iKW, which only allows finite formulae, by replacing the $\simeq_L$ rule of $L\bullet\mu$ by the following inference rule.*

$$\frac{}{\Gamma \vdash \bullet(\bullet A \to A) \to \bullet A}\ (W)$$

We observe that iKW is sound with respect to the Kripke semantics over $\bullet\mu$-frames, i.e., Definition 14, because the (W) rule is derivable in $L\bullet\mu$, by (approx) and (K), from the seemingly more general $(\bullet A \to A) \to A$, which is derivable

by the **Y**-combinator. And conversely, the axiom schema W : $\bullet(\bullet A \to A) \to \bullet A$ implies $(\bullet A \to A) \to A$ as follows:

$$\dfrac{\dfrac{}{\{\bullet A \to A\} \vdash \bullet A \to A}\ \text{(assump)} \qquad \dfrac{\dfrac{\dfrac{}{\{\bullet A \to A\} \vdash \bullet A \to A}\ \text{(assump)}}{\{\bullet A \to A\} \vdash \bullet(\bullet A \to A)}\ \text{(approx), (}\to\text{E)}}{\{\bullet A \to A\} \vdash \bullet A}\ \text{(W), (}\to\text{E)}}{\dfrac{\{\bullet A \to A\} \vdash A}{\{\} \vdash (\bullet A \to A) \to A}\ (\to\text{I})}\ (\to\text{E})$$

Then, since the only role of $(\simeq_L)$ for finite formulae in the proof of Lemma 1 is the derivability of $(\bullet D \to D) \to D$, which is used in the "if" part of the case $C = \bullet D$, we get the following.

Theorem 5 (Completeness of iKW). *The formal system iKW is also Kripke complete with respect to $\bullet\mu$-frames.*

And hence, by Theorem 3, $L\bullet\mu$ is a conservative extension of iKW.

6 Concluding Remarks

Two modal typing systems S-$\lambda\bullet\mu$ and F-$\lambda\bullet\mu$, which are respectively based on the simple and the F-semantics of types, and a formal system of the modal logic behind them have been presented. We have shown that the modal logic is Kripke complete with respect to intuitionistic, transitive and converse well-founded frames. The completeness also connects provability in the modal logic to type inhabitance in the two modal typing systems, and implies their decidability. We have also shown that the modal logic is a conservative extension of the intuitionistic version of the logic of provability. We have not, however, yet obtained corresponding results for the extended typing systems S-$\lambda\bullet\mu^+$ and F-$\lambda\bullet\mu^+$, which are also logically equivalent to each other, and completeness and decidability of typing and typability of λ-terms in all the typing systems presented in the present paper are also still open.

References

1. Nakano, H.: A modality for recursion. In: Proceedings of the 15th IEEE Symposium on Logic in Computer Science. IEEE Computer Society Press (2000) 255–266
2. Nakano, H.: A modality for recursion (technical report). Available as `http://www602.math.ryukoku.ac.jp/~nakano/papers/modality-tr01.ps` (2001)
3. Hindley, R.: The completeness theorem for typing λ-terms. Theoretical Computer Science **22** (1983) 1–17
4. Hindley, R.: Curry's type-rules are complete with respect to F-sematics too. Theoretical Computer Science **22** (1983) 127–133
5. Boolos, G.: The logic of provability. Cambridge University Press (1993)

6. Barendregt, H.P.: Lambda calculi with types. In Abramsky, S., Gabbay, D.M., Maibaum, T.S.E., eds.: Handbook of Logic in Computer Science. Volume 2. Oxford University Press (1992) 118–309
7. Cardone, F., Coppo, M.: Type inference with recursive types: syntax and semantics. Information and Computation **92** (1991) 48–80
8. Amadio, R.M., Cardelli, L.: Subtyping recursive types. ACM Transactions on Programming Languages and Systems **15** (1993) 575–631
9. Courcelle, B.: Fundamental properties of infinite trees. Theoretical Computer Science **25** (1983) 95–169

Appendix: Proof of Theorem 2

We first give alternative definitions of tail finiteness and positively (negatively) finiteness.

Definition 16. *Let V be a set of type variables. We define subsets $\mathbf{TF}^V$, $\mathbf{PF}$ and $\mathbf{NF}$ of $\mathbf{TExp}$ as follows:*

$$\begin{array}{rcl}
\mathbf{TF}^V & ::= & X \quad (X \notin V) \\
& | & \bullet\,\mathbf{TF}^V \;\mid\; \mathbf{TExp} \to \mathbf{TF}^V \;\mid\; \mu Y.\mathbf{TF}^{V\cup\{Y\}} \\
\mathbf{PF} & ::= & \mathbf{TVar} \;\mid\; \bullet\,\mathbf{PF} \;\mid\; \mathbf{NF} \to \mathbf{PF} \\
& | & \mu Y.A \quad (A \in \mathbf{TF}^{\{Y\}} \cap \mathbf{PF},\ \textit{and}\ Y \in ETV^-(A)\ \textit{implies}\ A \in \mathbf{NF}). \\
\mathbf{NF} & ::= & \mathbf{TVar} \;\mid\; \bullet\,\mathbf{NF} \;\mid\; \mathbf{PF} \to \mathbf{NF} \;\mid\; C \quad (C\ \textit{is an S-}\top\textit{-variant}) \\
& | & \mu Y.A \quad (A \in \mathbf{NF},\ \textit{and}\ Y \in ETV^-(A)\ \textit{implies}\ A \in \mathbf{TF}^{\{Y\}} \cap \mathbf{PF}).
\end{array}$$

Proposition 7. *(1) A is tail finite if and only if $A \in \mathbf{TF}^{\{\}}$.*
(2) A is positively finite if and only if $A \in \mathbf{PF}$.
(3) A is negatively finite if and only if $A \in \mathbf{NF}$.

It follows that tail finiteness and positively (negatively) finiteness are decidable properties of type expressions. Through these alternative definitions, we get the following proposition.

Proposition 8. *Suppose that $A \preceq B$.*

(1) If B is tail finite, then so is A.
(2) If B is positively finite, then so is A.
(3) If A is negatively finite, then so is B.

Proof of the first claim of Theorem 2. Suppose that $\Gamma \vdash M : A$ is derivable in S-$\lambda\bullet\mu^+$. We consider the frame $<\mathcal{N}, >>$, which consists of the set of non-negative integers and the "greater than" relation on it, over the term model $<\mathcal{V}, \cdot, [\![\ ]\!]>$ of untyped λ-calculus. Define a subset $\mathcal{K}$ of $\mathcal{V}$ as:

$$\mathcal{K} = \left\{ [\,x N_1 N_2 \ldots N_n\,] \;\middle|\; \begin{array}{l} x \text{ is an individual variable, } n \geq 0, \text{ and} \\ N_i \in \mathcal{V} \text{ for every } i \ (i = 1, 2, \ldots, n) \end{array} \right\}.$$

Taking ρ as $\rho(x) = [x]$ for any x, we get $[M] = [\![M]\!]_\rho \in \mathcal{I}_S(A)^\xi_p$ by Theorem 1 for S-$\lambda\bullet\mu^+$. Note also that $\mathcal{K} \subset \mathcal{I}_S(A)^\xi_p$ for any A and p by Definition 8.

Since ξ can be any type environment, it suffices to show that M has a head normal form whenever

(a) $A \in \mathbf{TF}^V$,
(b) $\xi(X)_p = \mathcal{K}$ for every p and $X \notin V$, and
(c) $[M] \in \mathcal{I}(A)^\xi_p$ for every p.

The proof proceeds by induction on the complexity of A, and by cases of the form of A. Suppose (a) through (c).

Case: $A = X$. In this case, $\mathcal{I}(A)^\xi_p = \xi(X)_p = \mathcal{K}$ by (a) and (b). Therefore, M obviously has a head normal form.

Case: $A = \bullet B$. In this case, $B \in \mathbf{TF}^V$ by (a). Therefore, M has a head normal form by the induction hypothesis. Note that (c) implies $[M] \in \mathcal{I}(B)^\xi_p$ for every p, because there exists some q such that $q \to p$.

Case: $A = B \to C$. In this case, $C \in \mathbf{TF}^V$ by (a). Let y be a fresh individual variable. Since $[M] \in \mathcal{I}(B \to C)^\xi_p$ and $[y] \in \mathcal{K} \subset \mathcal{I}(B)^\xi_p$ for every p, we get $[My] \in \mathcal{I}(C)^\xi_p$ for every p. Therefore, My has a head normal form, say L, by the induction hypothesis. There are two possible cases: for some K, (1) $M \xrightarrow[\beta]{*} K \in \mathcal{K}$ and $L = Ky \in \mathcal{K}$, or (2) $M \xrightarrow[\beta]{*} \lambda y.\, K$ and $K \xrightarrow[\beta]{*} L$. In either case, M has a head normal form.

Case: $A = \mu Y.B$. In this case, $B \in \mathbf{TF}^{V\cup\{Y\}}$ by (a). By Definition 8, we get $\mathcal{I}(\mu Y.B)^\xi_p = \mathcal{I}(B[\mu Y.B/Y])^\xi_p = \mathcal{I}(B)^{\xi'}_p$, where $\xi' = \xi[\mathcal{I}(\mu Y.B)^\xi_p/Y]$, since $\mathcal{I}(C[D/Y])^\xi_p = \mathcal{I}(C)^{\xi[\mathcal{I}(D)^\xi/Y]}_p$ holds for any C and D. Note that (a′) $B \in \mathbf{TF}^{V\cup\{Y\}}$, (b′) $\xi'(X)_p = \mathcal{K}$ for every p and $X \notin V \cup \{Y\}$, and (c′) $M \in \mathcal{I}(B)^{\xi'}_p$ for every p. Therefore, M has a head normal form by the induction hypothesis. □

As for the second claim of Theorem 2, we employ the following lemma.

Lemma 2. *Suppose that $A \not\preceq \top$. If $\Gamma \vdash xN_1N_2\ldots N_n : A$ is derivable in S-$\lambda\bullet\mu^+$, then $\Gamma(x) \preceq \bullet^{m_1}(B_1 \to \bullet^{m_2}(B_2 \to \ldots \to \bullet^{m_n}(B_n \to C)\ldots))$ for some $m_1, m_2, \ldots, m_n, B_1, B_2, \ldots, B_n$ and C such that*

1. *$\bullet^{m_1+m_2+\ldots+m_n}C \preceq A$, and*
2. *for every i $(0 \le i \le n)$, $\Gamma \vdash N_i \bullet^{m'_i} B_i$ is derivable for some m'_i.*

Proof. By induction on n. If $n = 0$, then since $A \not\preceq \top$, the derivation ends with:

$$\dfrac{\dfrac{}{\Gamma' \vdash x : \Gamma'(x)}\ (\text{var})}{\begin{array}{c}\vdots\ 0 \text{ or more } (\preceq)\text{'s}\\ \Gamma \vdash x : A\end{array}}$$

Therefore, we get $C \preceq A$ by taking C as $C = \Gamma'(x)$. If $n > 0$, then for some m', D and E, the derivation ends with:

$$\dfrac{\dfrac{\begin{array}{cc}\vdots & \vdots \\ \Gamma' \vdash x N_1 N_2 \ldots N_{n-1} : \bullet^{m'}(D \to E) & \Gamma' \vdash N_n : \bullet^{m'} D\end{array}}{\Gamma' \vdash x N_1 N_2 \ldots N_n : \bullet^{m'} E}\ (\to\mathrm{E})}{\begin{array}{c}\vdots \text{ 0 or more } (\preceq)\text{'s} \\ \Gamma \vdash x N_1 N_2 \ldots N_n : A\end{array}}$$

Note that $\bullet^{m'} E \preceq A$, and $E \not\simeq \top$ since $A \not\simeq \top$. By induction hypothesis, $\Gamma'(x) \preceq \bullet^{m_1}(B_1 \to \bullet^{m_2}(B_2 \to \ldots \to \bullet^{m_{n-1}}(B_{n-1} \to C') \ldots))$ for some m_1, m_2, …, m_{n-1}, B_1, B_2, …, B_{n-1} and C' such that:

- $\bullet^{m_1+m_2+\ldots+m_{n-1}} C' \preceq \bullet^{m'}(D \to E)$, and
- for every i $(0 \leq i \leq n-1)$, $\Gamma \vdash N_i \bullet^{m'_i} B_i$ is derivable for some m'_i.

This implies that there exist some m'', j, k, l, B_n and C such that:

- $\bullet^{m_1+m_2+\ldots+m_{n-1}} C' \simeq \bullet^{m''}(B_n \to C)$,
- $m''-j \leq m'-k$, and
- $\bullet^k D \preceq \bullet^{j+l} B_n$ and $\bullet^{j+l} C \preceq \bullet^k E$.

We then get $C' \preceq \bullet^{m_n}(B_n \to C)$, where $m_n = m''-m_1-m_2-\ldots-m_{n-1}$; and therefore, $\Gamma(x) \preceq \bullet^{m_1}(B_1 \to \bullet^{m_2}(B_2 \to \ldots \to \bullet^{m_{n-1}}(B_{n-1} \to \bullet^{m_n}(B_n \to C) \ldots))$. On the other hand, $\bullet^{m_1+m_2+\ldots+m_{n-1}+m_n} C = \bullet^{m''} C \preceq \bullet^{m'-k} C \preceq \bullet^{m'-k+j+l} C \preceq \bullet^{m'} E \preceq A$ and $\bullet^{m'} D \preceq \bullet^{m'-k+j+l} B_n$. We get the derivation of $\Gamma \vdash N_n : \bullet^{m'-k+j+l} B_n$ from the one of $\Gamma \vdash N_n : \bullet^{m'} D$ by $(\preceq)$. □

Proof of the second claim of Theorem 2. Suppose that $\Gamma \vdash M : A$ is derivable in S-$\lambda\bullet\mu^+$ for some A and Γ such that A is positively finite and $\Gamma(x)$ is negatively finite for every $x \in Dom(\Gamma)$. We show that for every n, every node of the Böhm-tree of M at the level n is head normalizable, by induction on n. Since A is positively finite, M is head normalizable by (1) of Theorem 2, that is

$$M \xrightarrow[\beta]{*} \lambda x_1.\, \lambda x_2.\, \ldots \lambda x_m.\, y N_1\, N_2\, \ldots\, N_l$$

for some x_1, x_2, …, x_m, y, N_1, N_2, …, N_l. By (3) of Proposition 2, $\Gamma \vdash \lambda x_1.\, \lambda x_2.\, \ldots \lambda x_m.\, y N_1\, N_2\, \ldots\, N_l : A$ is also derivable; and this implies that so is $\Gamma \cup \{x_1 : B_1, x_2 : B_2, \ldots, x_m : B_m\} \vdash y N_1\, N_2\, \ldots\, N_l : C$ for some B_1, B_2, …, B_m and C such that:

$$B_1 \to B_2 \to \ldots \to B_m \to C \preceq A.$$

Since A is positively finite, so is C, i.e., $C \not\simeq \top$, and $B_1, B_2, \ldots, B_m$ are negatively finite by Proposition 8. Let $\Gamma' = \Gamma \cup \{x_1 : B_1, x_2 : B_2, \ldots, x_m : B_m\}$. Since $C \not\simeq \top$, by Lemma 2, $\Gamma'(y) \preceq \bullet^{k_1}(D_1 \to \bullet^{k_2}(D_2 \to \ldots \to \bullet^{k_l}(D_l \to E) \ldots))$ for some k_1, k_2, …, k_l, D_1, D_2, …, D_l and E such that:

– $\bullet^{k_1+k_2+\ldots+k_l} E \preceq C$, and
– for every i $(0 \leq i \leq l)$, $\Gamma' \vdash N_i : \bullet^{k'_i} D_i$ is derivable for some k'_i.

Since C is positively finite, so is E, i.e., $E \not\simeq \top$; and therefore, D_i is positively finite for every i $(0 \leq i \leq l)$ because $\Gamma'(z)$ is negatively finite for every $z \in Dom(\Gamma')$. Therefore, by the induction hypothesis, for every i $(0 \leq i \leq l)$, every node of the Böhm-tree of N_i at a level less than n is head normalizable; that is, so is one of M at a level less than or equal to n. □