A Modality for Recursion ⋆
(Technical Report)⋆⋆

March 31, 2001

Hiroshi Nakano

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

Abstract. We propose a modal logic that enables us to handle self-referential formulae, including ones with negative self-references, which on one hand, would introduce a logical contradiction, namely Russell's paradox, in the conventional setting, while on the other hand, are necessary to capture a certain class of programs such as fixed point combinators and objects with so-called binary methods in object-oriented programming. Our logic provides a basis for axiomatic semantics of such a wider range of programs and a new framework for natural construction of recursive programs in the proofs-as-programs paradigm. We give the logic as a modal typing system with recursive types for the purpose of presentation, and show its soundness with respect to a realizability interpretation which implies the convergence of well-typed programs according to their types.

1 Introduction

Even though recursion, or self-reference, is an indispensable concept in both programs and their specifications, it is still far from obvious how to capture it in an axiomatic semantics such as the formulae-as-types notion of construction [17]. Only a rather restricted class of recursive programs (and specifications) has been captured in this direction as (co)inductive proofs over the (co)inductive data structures (see e.g., [9, 14, 24, 19, 27]), and, for example, negative self-references, which would be necessary to handle a certain range of programs such as fixed point combinators and objects with so-called binary methods in object-oriented programming, still remain out of the scope.

In this paper, we propose a modal logic that provides a basis for capturing such a wider range of programs in the proofs-as-programs paradigm. We give the logic as a modal typing system with recursive types for the purpose of presentation, and show its soundness with respect to a realizability interpretation which implies the convergence of well-typed programs according to their types.

Difficulty in binary-methods. Consider, for example, the specification $\mathbf{Nat}(n)$ of objects that represent a natural number n with a *method* which returns an object of $\mathbf{Nat}(n{+}m)$ when one of $\mathbf{Nat}(m)$ is given. It could be represented by a self-referential specification such as:

$$\mathbf{Nat}(n) \equiv ((n = 0) + (n > 0 \land \mathbf{Nat}(n{-}1)) \times (\forall m.\, \mathbf{Nat}(m) \to \mathbf{Nat}(n{+}m))),$$

where we assume that n and m range over the set of natural numbers; $+$, $\times$ and $\to$ are type constructors for direct sums, direct products and function spaces, respectively; $\land$ and $\forall$ have standard logical (annotative) meanings. Although it is not obvious whether this self-referential specification is meaningful in a certain mathematical sense, it could be a first approximation of the specification we want since this can be regarded as a refined version of recursive types which have been widely adopted as a basis for object-oriented type systems [1, 6]. At any rate, if we define an object $\mathbf{0}$ as:

$$\mathbf{0} \equiv <\mathbf{i_1} *, \lambda x.\, x>,$$

then it would satisfy $\mathbf{Nat}(0)$, where $\mathbf{i_1}$ is the injection into the first summand of direct sums and $*$ is a constant. We assume that any program satisfies annotative formulae such as $n = 0$ whenever they are true. We can easily define a function that satisfies $\forall n.\, \forall m.\, \mathbf{Nat}(n) \to \mathbf{Nat}(m) \to \mathbf{Nat}(n{+}m)$ as:

$$\mathbf{add}\; x\; y \equiv \mathbf{p_2}\; x\; y,$$

⋆ Research supported in part by 1999 Overseas Researcher Program of Ryukoku University.

⋆⋆ This report contains detailed proofs of results presented in [23] as well as additional ones.

or

$$\mathbf{add}'\, x\, y \equiv \mathbf{p_2}\, y\, x,$$

where $\mathbf{p_2}$ extracts second components, i.e., the method of addition in this particular case, from pairs. We could also define the successor function as a recursive program as:

$$\mathbf{s}\, x \equiv <\mathbf{i_2}\, x, \lambda y.\, \mathbf{add}\, x\, (\mathbf{s}\, y)>$$

or

$$\mathbf{s}'\, x \equiv <\mathbf{i_2}\, x, \lambda y.\, \mathbf{add}'\, x\, (\mathbf{s}'\, y)>.$$

In spite of the apparent symmetry between **add** and **add′**, which are both supposed to satisfy the same specification, the computational behaviors of **s** and **s′** are completely different. We can observe that **s** works as expected, but **s′** does not. For example, $\mathbf{p_2}\,(\mathbf{s}\,\mathbf{0})\,\mathbf{0}$ would be evaluated as: $\mathbf{p_2}\,(\mathbf{s}\,\mathbf{0})\,\mathbf{0} \to (\lambda y.\,\mathbf{add}\,\mathbf{0}\,(\mathbf{s}\,y))\,\mathbf{0} \to \mathbf{add}\,\mathbf{0}\,(\mathbf{s}\,\mathbf{0}) \to \mathbf{p_2}\,\mathbf{0}\,(\mathbf{s}\,\mathbf{0}) \to (\lambda x.\, x)\ (\mathbf{s}\ \mathbf{0}) \to \mathbf{s}\ \mathbf{0}$, whereas $\mathbf{p_2}\,(\mathbf{s}'\ \mathbf{0})\ \mathbf{0} \to (\lambda y.\ \mathbf{add}'\ \mathbf{0}\ (\mathbf{s}'\ y))\ \mathbf{0} \to \mathbf{add}'\ \mathbf{0}\ (\mathbf{s}'\ \mathbf{0}) \to \mathbf{p_2}\,(\mathbf{s}'\ \mathbf{0})\ \mathbf{0} \to \ldots$, and more generally, for any objects x and y of $\mathbf{Nat}(n)$ (for some n), $\mathbf{p_2}\,(\mathbf{s}'x)\,y \to \ldots \to \mathbf{p_2}\,(\mathbf{s}'y)\,x \to \ldots \to \mathbf{p_2}\,(\mathbf{s}'x)\,y \to \ldots.$

It should be noted that this sort of divergence would also be quite common in (careless) recursive definitions of programs even if we did not have to handle object-oriented specifications like $\mathbf{Nat}(n)$. The peculiarity here is the fact that the divergence is caused by a program, **add′**, which is supposed to satisfy the same specification as **add**. This example shows such a loss of the compositionality of programs with respect to the specifications that imply their termination, or convergence. It also suggests that, to overcome this difficulty, **add** and **add′** should have different specifications, and accordingly the definition of $\mathbf{Nat}(n)$ should be revised in some way in order to force it.

$\lambda\mu$ and its logical inconsistency. The typing system $\lambda\mu$ (see [4], and Section 3 of the present paper for a summary) is a simply-typed lambda calculus with recursive types, where any form of self-references, including negative ones, is permitted. A non-trivial model for such unrestricted recursive types was developed by MacQueen, Plotkin and Sethi [22], and has been widely adopted as a theoretical basis for object-oriented type systems [1, 6].

On the other hand, it is well known that logical formulae with such unrestricted self-references would introduce a contradiction (variant of Russell's paradox). Therefore, logical systems must have certain restrictions on the forms of self-references (if ever allowed) in order to keep themselves sound; for example, μ-calculus [25, 20] does not allow negative self-references (see also [13]).

Through the formulae-as-types notion, this paradox corresponds to the fact that every type of $\lambda\mu$ is inhabited by a diverging program which does not produce any information; for example, the λ-term $(\lambda x.\, xx)(\lambda x.\, xx)$ can be typed with every type in $\lambda\mu$. Therefore, even with the model mentioned above, types can be regarded only as partial specifications of programs, and that is considered the reason why we lost the compositionality of programs in the $\mathbf{Nat}(n)$ case, where we regarded convergence of programs as a part of their specifications. This shows a contrast with the success of $\lambda\mu$ as a basis for type systems of object-oriented program languages, where the primary purpose of types, i.e., coarse specifications, is to prevent run-time type errors, and termination of programs is out of the scope.

The logical inconsistency of $\lambda\mu$ also implies that no mater how much types, or specifications, are refined, convergence of programs can not be expressed by them, and must be handled by endowing the typing system with some facilities for discussing computational properties of programs. For example, Constable et al. adopted this approach in their pioneering works to incorporate recursive definitions and partial objects into constructive type theory [10, 11]. However, in this paper, we will pursue another approach such that types themselves can express convergence of programs.

Towards the approximation modality. Suppose that we have a recursive program f defined by:

$$f \equiv F(f),$$

and want to show that f satisfies a certain specification S. Since the denotational meaning of f is given as the least fixed point of F, i.e., $f = \sup_{n<\omega} F^n(\bot)$, a possible way to do that would be to apply Scott's fixed point induction [26] by showing that:

- $\bot$ satisfies S,
- $F(x)$ satisfies S provided that x satisfies S, and
- S is chain closed.

However, this does not suffice for our purpose if S includes some requirement about the convergence of f, because obviously $\perp$, or even $F^n(\perp)$, could not satisfy the requirement. So we need more refined approach. The failure of the naive fixed point induction above suggests that the specification to be satisfied by each $F^n(\perp)$ inherently depends on n, and the requirement concerning its convergence must become stronger when n increases. This leads us to a layered version of the fixed-point induction scheme as follows: in order to show that f satisfies S, it suffices to find an infinite sequence $S_0, S_1, S_2, \ldots$ of properties, or (virtual) specifications, such that:

(1) $S = \bigcap_{n<\omega} S_n$,
(2) $S_{n+1} \subset S_n$,
(3) $\perp$ satisfies S_0,
(4) $F(x)$ satisfies S_{n+1} provided that x satisfies S_n, and
(5) S_n is chain closed.

For, since $F^n(\perp) \in S_n$ for every n by (3) and (4), we get $F^k(\perp) \in S_n$ for every $k \geq n$ by (2). This and (5) imply $f \in S_n$ for every n, and consequently $f \in S$ by (1).

In this scheme, the sequence $S_0, S_1, S_2, \ldots$ can be regarded as a successive approximation of S, and F a (higher-order) program which constructs a program that satisfies S_{n+1} from one that satisfies S_n. It should be also noted that F works independently of n. This uniformity of F over n leads us to consider a formalization of this scheme in a modal logic, where the set of possible worlds (in the sense of Kripke semantics) consists of all non-negative integers, and S_n in the induction scheme above corresponds to the interpretation of S in the world n. We now write $x\ \mathbf{r}_k\ S$ to denote the fact that x satisfies the interpretation of S in the world k, and define a modality, say $\bullet$, as:

$$x\ \mathbf{r}_k\ \bullet S \quad \text{iff} \quad k = 0\ \text{ or }\ x\ \mathbf{r}_{k-1}\ S.$$

The condition (2) of the induction scheme says that $x\ \mathbf{r}_k\ S$ implies $x\ \mathbf{r}_l\ S$ for every $l \leq k$; in other words, the interpretation of specifications should be *hereditary* with respect to the accessibility relation $>$. In such a modal framework, the specification to be satisfied by F can be represented by $\bullet S \to S$ provided that the $\to$-connective is interpreted in the standard way in each world, and our induction scheme can be rewritten as:

if $\perp\ \mathbf{r}_0\ S$ and $F\ \mathbf{r}_k\ \bullet S \to S$ for every $k > 0$, then $f\ \mathbf{r}_k\ S$ for every k.

Furthermore, if we assume that S_0 is a trivial specification which is satisfiable by any program, then, shifting the possible worlds downwards by one, we can simplify this to:

$(*)$ if $F\ \mathbf{r}_k\ \bullet S \to S$ for every k, then $f\ \mathbf{r}_k\ S$ for every k.

Although this assumption about S_0 somewhat restricts our choice of the sequence $S_0, S_1, S_2, \ldots$, it could be thought rather reasonable because, at any rate, S_0 must be an almost trivial specification that is even satisfiable by $\perp$. Note that S_{n+1} occurring in the induction now corresponds to the interpretation of S in the world n, and S_0 corresponds to the interpretation of $\bullet S$ in the world 0.

From this interpretation, we can extract some fundamental properties concerning the $\bullet$-modality, which introduce a subsumption, or subtyping, relation over specifications into our modal framework. First, the hereditary interpretation of specifications implies the following property:

– $x\ \mathbf{r}_k\ S$ implies $x\ \mathbf{r}_k\ \bullet S$.

Second, this and the standard interpretations of $\to$ imply the following two properties:

– $x\ \mathbf{r}_k\ S \to T$ implies $x\ \mathbf{r}_k\ \bullet S \to \bullet T$, and
– $x\ \mathbf{r}_k\ \bullet S \to \bullet T$ implies $x\ \mathbf{r}_k\ \bullet(S \to T)$.

Furthermore, if $x\ \mathbf{r}_k\ \top \to \top$ for every x and k, where $\top$ is the trivial specification which is satisfiable by any program, i.e., the universe of (meanings of) programs, then the converse of the second one is also true, that is:

– $x\ \mathbf{r}_k\ \bullet(S \to T)$ implies $x\ \mathbf{r}_k\ \bullet S \to \bullet T$.

Note that this is not always the case because we could consider non-extensional interpretations, e.g., F-semantics [15], in which $\lambda x.\perp\ \mathbf{r}_k\ \top \to \top$ holds, but $\perp\ \mathbf{r}_k\ \top \to \top$ does not.

Specification-level self-references. This modal framework introduced for program-level self-references also provides a basis for specification-level self-references. Suppose that we have a self-referential specification such as:

$$S = \phi(S).$$

As we saw in the $\mathbf{Nat}(n)$ case, negative reference to S in ϕ can introduce a contradiction in the conventional setting, and this is still true in our modal framework. However, in the world n, we can now refer to the interpretation of S in any world $k < n$ without worrying about the contradiction. That is, as long as S occurs only in scopes of the modal operator $\bullet$ in ϕ, the interpretation of S is well-defined and given as a fixed point of ϕ, which is actually shown to be unique. For example, if S is defined as $S = \bullet S \to T$, then S could be interpreted in each world as follows:

$$\begin{aligned}
S_0 &= \top \to T_0 \\
S_1 &= S_0 \cap ((\top \to T_0) \to T_1) \\
S_2 &= S_1 \cap ((S_0 \cap ((\top \to T_0) \to T_1)) \to T_2) \\
&\;\vdots \\
S_{n+1} &= S_n \cap (S_n \to T_{n+1}) \\
&\;\vdots
\end{aligned}$$

where S_k and T_k are the interpretations of S and T in the world k, respectively, and the notations such as $\top$ and $\to$ are abused to denote their expected interpretations also. Note that this kind of self-references provides us a method to define the sequence $S_0,\ S_1,\ S_2,\ \ldots$ for the refined induction scheme when we derive properties of recursive programs, and the induction scheme would be useless if we did not have such a method.

In the following sections, we will see that this form of specification-level self-references is quite powerful, and captures a wide range of specifications including those which are not representable in the conventional setting such as ones for **add** and **add′** in the $\mathbf{Nat}(n)$ case. Furthermore, the modal version $(*)$ of the induction scheme turns out to be derivable from other properties of the $\bullet$-modality and such self-referential specifications, where the derivation corresponds to fixed point combinators, such as Curry's **Y**. This also gives us a way to construct recursive programs based on the proofs-as-programs notion.

2 Untyped λ-calculus

We begin with the definition of the standard untyped λ-calculus with individual constants.

Definition 2.1 (Untyped λ-terms). The syntax of the λ-terms is defined relatively to the following two disjoint sets: **Const** of *individual constants* $(c, d, \ldots)$ and **Var** of countably infinite *individual variables* $(f, g, h, x, y, z, \ldots)$. The set **Exp** of *$\lambda$-terms* is defined by the following BNF notation:

$$\begin{array}{lcll}
\mathbf{Exp} & ::= & \mathbf{Const} & \text{(individual constants)} \\
 & \mid & \mathbf{Var} & \text{(individual variables)} \\
 & \mid & \lambda\mathbf{Var}.\,\mathbf{Exp} & \text{(λ-abstractions)} \\
 & \mid & \mathbf{Exp}\mathbf{Exp} & \text{(applications).}
\end{array}$$

We use $M, N, K, L, \ldots$ to denote λ-terms. Free and bound occurrences of individual variables and the notion of α-convertibility are defined in the standard manner. Hereafter, we identify λ-terms by this α-convertibility. We denote the set of individual variables occurring freely in M by $FV(M)$, and use $M[N_1/x_1, \ldots, N_n/x_n]$ to denote the λ-term obtained from a λ-term M by substituting $N_1, \ldots, N_n$ for each free occurrence of individual variables $x_1, \ldots, x_n$, respectively, with necessary α-conversion to avoid accidental capture of free variables.

Definition 2.2 (β-reduction). The standard notion of β-reduction, a binary relation $\underset{\beta}{\to}$ over **Exp**, is defined as follows:

$$\mathcal{C}[(\lambda x.\, M\,)N] \underset{\beta}{\to} \mathcal{C}[M[N/x]],$$

where $\mathcal{C}$ is an arbitrary context of λ-term.

We denote the transitive and reflexive closure of $\underset{\beta}{\to}$ by $\overset{*}{\underset{\beta}{\to}}$, and the symmetric closure of $\underset{\beta}{\to}$ by $\underset{\beta}{\leftrightarrow}$. We define the equivalence relation $\underset{\beta}{=}$ as the transitive and reflexive closure of $\underset{\beta}{\leftrightarrow}$.

Definition 2.3. Let ρ be a mapping from a set T to a set S, and let $x \in T$ and $v \in S$. We define a mapping $\rho[v/x]$ by:

$$\rho[v/x](y) = \begin{cases} v & (y = x) \\ \rho(y) & (y \neq x) \end{cases}$$

Our intended semantics for untyped λ-terms is summarized as the following, where we do not require extensionality with respect to their interpretations.

Definition 2.4 **(β-model).** A β-model of **Exp** is a tuple $<\mathcal{V},\ \cdot,\ \sigma,\ [\![\ ]\!]^{\mathcal{V}}>$ such that:

1. $\mathcal{V}$: a non-empty set.
2. σ : **Const** $\rightarrow \mathcal{V}$.
3. $- \cdot - : \mathcal{V} \times \mathcal{V} \rightarrow \mathcal{V}$.
4. $[\![-]\!]^{\mathcal{V}}_{-}$: **Exp** $\rightarrow$ (**Var** $\rightarrow \mathcal{V}) \rightarrow \mathcal{V}$.
5. $[\![x]\!]^{\mathcal{V}}_{\rho} = \rho(x)$.
6. $[\![c]\!]^{\mathcal{V}}_{\rho} = \sigma(c)$.
7. $[\![MN]\!]^{\mathcal{V}}_{\rho} = [\![M]\!]^{\mathcal{V}}_{\rho} \cdot [\![N]\!]^{\mathcal{V}}_{\rho}$.
8. $[\![\lambda x.\, M]\!]^{\mathcal{V}}_{\rho} \cdot v = [\![M]\!]^{\mathcal{V}}_{\rho[v/x]}$.
9. If $M \underset{\beta}{=} N$, then $[\![M]\!]^{\mathcal{V}}_{\rho} = [\![N]\!]^{\mathcal{V}}_{\rho}$.

3 A brief review of $\lambda\mu$

In this section, we give a brief review of the typing system $\lambda\mu$, which is a simply typed lambda calculus with recursive types.

Definition 3.1 **(Type expressions of $\lambda\mu$).** The syntax of the type expressions of $\lambda\mu$ is defined relatively to the following two sets: **TConst** of *type constants* $(P, Q, R, \ldots)$ and **TVar** of countably infinite *type variables* $(X, Y, Z, \ldots)$. The set $\mathbf{TExp}_{\lambda\mu}$ of *type expressions* of $\lambda\mu$ is defined as follows:

$$\begin{array}{rcll} \mathbf{TExp}_{\lambda\mu} & ::= & \mathbf{TConst} & \text{(type constants)} \\ & | & \mathbf{TVar} & \text{(type variables)} \\ & | & \mathbf{TExp}_{\lambda\mu} \rightarrow \mathbf{TExp}_{\lambda\mu} & \text{(function types)} \\ & | & \mu\mathbf{TVar}.\ \mathbf{TExp}_{\lambda\mu} & \text{(recursive types).} \end{array}$$

We use $A, B, C, D, \ldots$ to denote type expressions of $\lambda\mu$. We regard a type variable X as bound in $\mu X.A$. We 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. We denote the set of type variables occurring freely in A by $FTV(A)$. We regard α-convertible type expressions as identical; for example, $\mu X.X \rightarrow Y = \mu Z.Z \rightarrow Y$. We use $\mu\boldsymbol{X}A$ to denote a type expression of the form $\mu X_1.\mu X_2. \ldots \mu X_n.A$, where n is a non-negative integer.

Definition 3.2. A type expression A of $\lambda\mu$ is $\lambda\mu$-*proper*[1] in X if and only if X occurs freely only in scopes of the $\rightarrow$-operator in A. That is,

1. A type constant P is always $\lambda\mu$-proper in X.
2. A type variable Y is $\lambda\mu$-proper in X if and only if $Y \neq X$.
3. $A \rightarrow B$ is always $\lambda\mu$-proper in X.
4. $\mu Y.A$ is $\lambda\mu$-proper in X if and only if so is A or $Y = X$.

Note that A is $\lambda\mu$-proper in X if and only if for every $\boldsymbol{Y}$, $A \neq \mu\boldsymbol{Y}.X$.

Definition 3.3. We define the equivalence relation $\simeq_{\lambda\mu}$ over $\mathbf{TExp}_{\lambda\mu}$ as the smallest binary relation that satisfies:

($\simeq_{\lambda\mu}$-reflex)	$A \simeq_{\lambda\mu} A$.
($\simeq_{\lambda\mu}$-symm)	If $A \simeq_{\lambda\mu} B$, then $B \simeq_{\lambda\mu} A$.
($\simeq_{\lambda\mu}$-trans)	If $A \simeq_{\lambda\mu} B$ and $B \simeq_{\lambda\mu} C$, then $A \simeq_{\lambda\mu} C$.

[1] Many authors say *contractive*.

($\simeq_{\lambda\mu}$-$\to$)	If $A \simeq_{\lambda\mu} C$ and $B \simeq_{\lambda\mu} D$, then $A \to B \simeq_{\lambda\mu} C \to D$.
($\simeq_{\lambda\mu}$-fix)	$\mu X.A \simeq_{\lambda\mu} A[\mu X.A/X]$.
($\simeq_{\lambda\mu}$-uniq)	If $A \simeq_{\lambda\mu} C[A/X]$ and C is $\lambda\mu$-proper in X, then $A \simeq_{\lambda\mu} \mu X.C$.

Intuitively, two type expressions of $\lambda\mu$ are equivalent modulo $\simeq_{\lambda\mu}$, if they have the same (possibly infinite) type expression obtained by unfolding recursive types $\mu X.A$ occurring in them to $A[\mu X.A/X]$ indefinitely. This equivalence relation is known to be decidable (see [8] and [3]).

Proposition 3.4. *Let n be a non-negative integer, X_1, X_2, ..., X_n type variables, and A, B_1, B_2, ..., B_n, C_1, C_2, ..., C_n type expressions of $\lambda\mu$. If $B_i \simeq_{\lambda\mu} C_i$ for every i ($i = 1, 2, \ldots, n$), then $A[B_1/X_1, B_2/X_2, \ldots, B_n/X_n] \simeq_{\lambda\mu} A[B_1/X_1, B_2/X_2, \ldots, B_n/X_n]$.*

Proof. By induction on the structure of A, and by cases of the form of A. The only interesting case is when $A = \mu Y.A'$ for some Y and A'. We can assume that $Y \notin \{X_i\} \cup FTV(B_i) \cup FTV(C_i)$ for every i. In the sequel, we use abbreviations $[\boldsymbol{B}/\boldsymbol{X}]$ and $[\boldsymbol{C}/\boldsymbol{X}]$ for $[B_1/X_1, B_2/X_2, \ldots, B_n/X_n]$ and $[C_1/X_1, C_2/X_2, \ldots, C_n/X_n]$, respectively. By the induction hypothesis, we have

$$\begin{aligned}
\mu Y.A'[\boldsymbol{B}/\boldsymbol{X}] &\simeq_{\lambda\mu} A'[\boldsymbol{B}/\boldsymbol{X}][\mu Y.A'[\boldsymbol{B}/\boldsymbol{X}]/Y] && \text{(by } (\simeq_{\lambda\mu}\text{-fix}))\\
&= A'[\boldsymbol{B}/\boldsymbol{X},\, \mu Y.A'[\boldsymbol{B}/\boldsymbol{X}]/Y] && \text{(since } Y \notin FTV(\boldsymbol{B}))\\
&\simeq_{\lambda\mu} A'[\boldsymbol{C}/\boldsymbol{X},\, \mu Y.A'[\boldsymbol{B}/\boldsymbol{X}]/Y] && \text{(by the ind. hyp.)}\\
&= A'[\boldsymbol{C}/\boldsymbol{X}][\mu Y.A'[\boldsymbol{B}/\boldsymbol{X}]/Y] && \text{(since } Y \notin FTV(\boldsymbol{C}))
\end{aligned}$$

Since $A'[\boldsymbol{C}/\boldsymbol{X}]$ is also $\lambda\mu$-proper in Y, we now get $\mu Y.A'[\boldsymbol{B}/\boldsymbol{X}] \simeq_{\lambda\mu} \mu Y.A'[\boldsymbol{C}/\boldsymbol{X}]$ by ($\simeq_{\lambda\mu}$-uniq); and therefore, $A[\boldsymbol{B}/\boldsymbol{X}] \simeq_{\lambda\mu} A[\boldsymbol{C}/\boldsymbol{X}]$. □

Proposition 3.5. *Let n be a non-negative integer, X_1, X_2, ..., X_n type variables, and A, B, C_1, C_2, ..., C_n, D_1, D_2, ..., D_n, type expressions of $\lambda\mu$. If $A \simeq_{\lambda\mu} B$ and $C_i \simeq_{\lambda\mu} D_i$ for every i ($i = 1, 2, \ldots, n$), then $A[C_1/X_1, C_2/X_2, \ldots, C_n/X_n] \simeq_{\lambda\mu} B[D_1/X_1, D_2/X_2, \ldots, D_n/X_n]$.*

Proof. By induction on the derivation of $A \simeq_{\lambda\mu} B$, and by cases of the last rules applied in the derivation. Use Proposition 3.4 in case of ($\simeq_{\lambda\mu}$-reflex). If the last rule is ($\simeq_{\lambda\mu}$-fix), then $A = \mu Y.A'$ and $B = A'[\mu Y.A'/Y]$ for some Y and A'. We can assume that $Y \notin \{X_i\} \cup FTV(C_i) \cup FTV(D_i)$ for every i.

$$\begin{aligned}
\mu Y.A'[\boldsymbol{C}/\boldsymbol{X}] &\simeq_{\lambda\mu} A'[\boldsymbol{C}/\boldsymbol{X}][\mu Y.A'[\boldsymbol{C}/\boldsymbol{X}]/Y] && \text{(by } (\simeq_{\lambda\mu}\text{-fix}))\\
&\simeq_{\lambda\mu} A'[\boldsymbol{C}/\boldsymbol{X},\, \mu Y.A'[\boldsymbol{C}/\boldsymbol{X}]/Y] && \text{(since } Y \notin FTV(\boldsymbol{C}))\\
&\simeq_{\lambda\mu} A'[\boldsymbol{D}/\boldsymbol{X},\, \mu Y.A'[\boldsymbol{C}/\boldsymbol{X}]/Y] && \text{(by Proposition 3.4)}\\
&\simeq_{\lambda\mu} A'[\boldsymbol{D}/\boldsymbol{X}][\mu Y.A'[\boldsymbol{C}/\boldsymbol{X}]/Y] && \text{(since } Y \notin FTV(\boldsymbol{D}))
\end{aligned}$$

Since $A'[\boldsymbol{D}/\boldsymbol{X}]$ is also $\lambda\mu$-proper in Y, we now get $\mu Y.A'[\boldsymbol{C}/\boldsymbol{X}] \simeq_{\lambda\mu} \mu Y.A'[\boldsymbol{D}/\boldsymbol{X}]$ by ($\simeq_{\lambda\mu}$-uniq); and therefore,

$$\begin{aligned}
A[\boldsymbol{B}/\boldsymbol{X}] &= \mu Y.A'[\boldsymbol{C}/\boldsymbol{X}]\\
&\simeq_{\lambda\mu} \mu Y.A'[\boldsymbol{D}/\boldsymbol{X}]\\
&\simeq_{\lambda\mu} A'[\boldsymbol{D}/\boldsymbol{X}][\mu Y.A'[\boldsymbol{D}/\boldsymbol{X}]/Y] && \text{(by } (\simeq_{\lambda\mu}\text{-fix}))\\
&= A'[\boldsymbol{D}/\boldsymbol{X},\, \mu Y.A'[\boldsymbol{D}/\boldsymbol{X}]/Y] && \text{(since } Y \notin FTV(\boldsymbol{D}))\\
&= A'[\mu Y.A'/Y][\boldsymbol{D}/\boldsymbol{X}]\\
&= B[\boldsymbol{D}/\boldsymbol{X}].
\end{aligned}$$

If the last rule is ($\simeq_{\lambda\mu}$-uniq), then $B = \mu Y.B'$ for some Y and B' such that $A \simeq_{\lambda\mu} B'[A/Y]$ and B' is $\lambda\mu$-proper in Y. We can assume that $Y \notin \{X_i\} \cup FTV(C_i) \cup FTV(D_i)$ for every i.

$$\begin{aligned}
A[\boldsymbol{C}/\boldsymbol{X}] &\simeq_{\lambda\mu} B'[A/Y][\boldsymbol{D}/\boldsymbol{X}] && \text{(by the ind. hyp.)}\\
&= B'[\boldsymbol{D}/\boldsymbol{X},\, A[\boldsymbol{D}/\boldsymbol{X}]/Y]\\
&= B'[\boldsymbol{D}/\boldsymbol{X}][A[\boldsymbol{D}/\boldsymbol{X}]/Y] && \text{(since } Y \notin FTV(\boldsymbol{D}))\\
&\simeq_{\lambda\mu} B'[\boldsymbol{D}/\boldsymbol{X}][A[\boldsymbol{C}/\boldsymbol{X}]/Y] && \text{(by Proposition 3.4)}
\end{aligned}$$

Since $B'[\boldsymbol{D}/\boldsymbol{X}]$ is also $\lambda\mu$-proper in Y, we now get $A[\boldsymbol{C}/\boldsymbol{X}] \simeq_{\lambda\mu} \mu Y.B'[\boldsymbol{D}/\boldsymbol{X}]$ by ($\simeq_{\lambda\mu}$-uniq); and therefore, $A[\boldsymbol{C}/\boldsymbol{X}] \simeq_{\lambda\mu} B[\boldsymbol{D}/\boldsymbol{X}]$. □

Proposition 3.6. *Suppose that A and B are both $\lambda\mu$-proper in X. If $A \simeq B$, then $\mu X.A \simeq \mu X.B$.*

Proof. We get $\mu X.A \simeq_{\lambda\mu} A[\mu X.A/X] \simeq_{\lambda\mu} B[\mu X.A/X]$ by ($\simeq_{\lambda\mu}$-fix) and Proposition 3.5. Therefore, since B is $\lambda\mu$-proper in X, we get $\mu X.A \simeq_{\lambda\mu} \mu X.B$ by ($\simeq_{\lambda\mu}$-uniq). □

Definition 3.7 (Typing contexts). A *typing context*, or a *context* for short, of $\lambda\mu$ is a finite mapping that assigns a type expression of $\lambda\mu$ to each individual variable of its domain. We use Γ, Γ', ... to denote contexts, and $\{x_1 : A_1, \dots, x_m : A_m\}$ to denote a context whose domain is $\{x_1, \dots, x_m\}$ and that assigns A_i to x_i for every i, where $A_1, \dots, A_m$ are type expressions of $\lambda\mu$, and $x_1, \dots, x_m$ are distinct individual variables.

Definition 3.8 ($\lambda\mu$). Let τ be a mapping that assigns a type constant $\tau(c)$ to each individual constant c. The typing system $\lambda\mu$ is defined relatively to this τ by the following derivation rules:

$$\frac{}{\Gamma \cup \{x : A\} \vdash_{\lambda\mu} x : A}\ (\text{var}) \qquad \frac{}{\Gamma \vdash_{\lambda\mu} c : \tau(c)}\ (\text{const}) \qquad \frac{\Gamma \vdash_{\lambda\mu} M : A}{\Gamma \vdash_{\lambda\mu} M : A'}\ (\simeq_{\lambda\mu}) \quad (A \simeq_{\lambda\mu} A')$$

$$\frac{\Gamma \cup \{x : A\} \vdash_{\lambda\mu} M : B}{\Gamma \vdash_{\lambda\mu} \lambda x.\, M : A \to B}\ (\to\mathrm{I}_{\lambda\mu}) \qquad \frac{\Gamma_1 \vdash_{\lambda\mu} M : A \to B \quad \Gamma_2 \vdash_{\lambda\mu} N : A}{\Gamma_1 \cup \Gamma_2 \vdash_{\lambda\mu} MN : B}\ (\to\mathrm{E}_{\lambda\mu})$$

Definition 3.9 (Realizability models of $\lambda\mu$). A realizability model of $\lambda\mu$ is a tuple $<\mathcal{V}, \cdot, \sigma, [\![\ ]\!]^{\mathcal{V}}, \mathcal{T}, \delta, [\![\ ]\!]^{\mathcal{T}}>$ such that:

1. $<\mathcal{V}, \cdot, \sigma, [\![\ ]\!]^{\mathcal{V}}>$ is a β-model of **Exp**.
2. $\mathcal{T} \subset \mathcal{P}(\mathcal{V}) \quad (= \{\, S \mid S \subset \mathcal{V} \,\})$
3. $\delta : \textbf{TConst} \to \mathcal{T}$
4. $\sigma(c) \in \delta(\tau(c))$
5. $[\![-]\!]^{\mathcal{T}}_{-} : \textbf{TExp}_{\lambda\mu} \to (\textbf{TVar} \to \mathcal{T}) \to \mathcal{T}$
6. $[\![X]\!]^{\mathcal{T}}_{\eta} = \eta(X)$
7. $[\![P]\!]^{\mathcal{T}}_{\eta} = \delta(P)$
8. $[\![A \to B]\!]^{\mathcal{T}}_{\eta} = \{\, v \mid v \cdot u \in [\![B]\!]^{\mathcal{T}}_{\eta}$ for every $u \in [\![A]\!]^{\mathcal{T}}_{\eta} \,\}$
9. If $A \simeq_{\lambda\mu} B$, then $[\![A]\!]^{\mathcal{T}}_{\eta} = [\![B]\!]^{\mathcal{T}}_{\eta}$.

It is not straightforward to construct a non-trivial realizability model of $\lambda\mu$. The first non-trivial model was developed by MacQueen, Plotkin and Sethi [22], by constructing a complete metric space of types and by interpreting recursive types as the fixed points of contractive type constructors (see also [2, 8]).

Proposition 3.10 (Soundness of $\lambda\mu$). *Let $<\mathcal{V}, \cdot, \sigma, [\![\ ]\!]^{\mathcal{V}}, \mathcal{T}, \delta, [\![\ ]\!]^{\mathcal{T}}>$ be a realizability model of $\lambda\mu$. If $\{x_1 : A_1, \dots, x_n : A_n\} \vdash_{\lambda\mu} M : B$ is derivable, then $[\![M]\!]^{\mathcal{V}}_{\rho} \in [\![B]\!]^{\mathcal{T}}_{\eta}$ for every η and ρ provided $\rho(x_i) \in [\![A_i]\!]^{\mathcal{T}}_{\eta}$ ($i = 1, 2, \dots, n$).*

Proof. By induction on the derivation and by cases of the last rule used in the derivation. □

Nevertheless, as mentioned in Introduction, unrestricted self-references allowed in $\lambda\mu$ cause a logical contradiction as follows (Curry's paradox):

Proposition 3.11. $\vdash_{\lambda\mu} (\lambda x.\, xx)\,(\lambda x.\, xx) : A$ *is derivable for any type expression A.*

Proof. Let $C = \mu X.X \to A$, and Π as follows:

$$\Pi = \cfrac{\cfrac{\cfrac{x : C \vdash_{\lambda\mu} x : C}{x : C \vdash_{\lambda\mu} x : C \to A}\ (\simeq_{\lambda\mu}) \quad x : C \vdash_{\lambda\mu} x : C}{x : C \vdash_{\lambda\mu} xx : A}\ (\to E)}{\vdash_{\lambda\mu} \lambda x.\, xx : C \to A}\ (\to I)$$

Then, we can derive it as follows:

$$\cfrac{\begin{array}{c}\vdots\ \Pi \\ \vdash_{\lambda\mu} \lambda x.\, xx : C \to A\end{array} \quad \cfrac{\begin{array}{c}\vdots\ \Pi \\ \vdash_{\lambda\mu} \lambda x.\, xx : C \to A\end{array}}{\vdash_{\lambda\mu} \lambda x.\, xx : C}\ (\simeq_{\lambda\mu})}{\vdash_{\lambda\mu} (\lambda x.\, xx)\,(\lambda x.\, xx) : A}\ (\to E)$$

□

4 The typing system $\lambda\bullet\mu$

We now define a modal typing system, which is denoted by $\lambda\bullet\mu$, based on the idea discussed in Introduction. First, as a preparation for introducing the syntax of type expressions, we give the one of pseudo type expressions, which are obtained by adding a unary type constructor $\bullet$ to the one of $\mathbf{TExp}_{\lambda\mu}$.

Definition 4.1. We define the set **PTExp** of *pseudo type expressions* as follows:

$$\begin{array}{rlll} \mathbf{PTExp} & ::= & \mathbf{TConst} & \text{(type constants)} \\ & \mid & \mathbf{TVar} & \text{(type variables)} \\ & \mid & \mathbf{PTExp} \to \mathbf{PTExp} & \text{(function types)} \\ & \mid & \bullet\mathbf{PTExp} & \text{(approximative types)} \\ & \mid & \mu\mathbf{TVar}.\mathbf{PTExp} & \text{(recursive types).} \end{array}$$

We assume that $\to$ associates to the right as usual, and each (pseudo) type constructor associates according to the following priority:

$$\text{(Low)} \qquad \mu X. \quad < \quad \to \quad < \quad \bullet \qquad \text{(High)}.$$

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 4.2 **($\top$-variants).** A type expression A is a $\top$-*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 $m_i + m_{i+1} + m_{i+2} + \ldots + m_n \geq 1$.

Definition 4.3 **(Properness).** A pseudo type expression A is *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 a $\top$-variant. In other words:

1. A type constant P is always proper in X.
2. A type variable Y is proper in X if and only if $Y \not\equiv X$.
3. $\bullet A$ is always proper in X.
4. $A \to B$ is proper in X if and only if (a) so are both A and B, or (b) B is a $\top$-variant.
5. $\mu Y.A$ is proper in X if and only if so is A or $Y = X$.

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

Definition 4.4 **(Type expressions of $\lambda\bullet\mu$).** We define the set **TExp** of *type expressions* as the smallest set of pseudo type expressions that satisfies:

1. $P \in$ **TExp** for every type constant P.
2. $X \in$ **TExp** for every type variable X.
3. If $A \in$ **TExp** and $B \in$ **TExp**, then $A \to B \in$ **TExp**.
4. If $A \in$ **TExp**, then $\bullet A \in$ **TExp**.
5. If $A \in$ **TExp** and A is proper in X, then $\mu X.A \in$ **TExp**.

In other words, a type expression is a pseudo type expression such that A is proper in X for any of its subexpressions in the form of $\mu X.A$. We denote the set of type expressions of $\lambda\bullet\mu$ by **TExp**.

For example, P, 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 of $\lambda\bullet\mu$, and neither $\mu X.X \to Y$ nor $\mu X.\mu Y.X \to Y$ is a type expression of $\lambda\bullet\mu$. We also use $A, B, C, D, \ldots$ to denote type expressions of $\lambda\bullet\mu$, and define other notations such as $FTV(A)$ and $A[B_1/X_1, \ldots, B_n/X_n]$ similarly to the case of $\lambda\mu$.

Definition 4.5. Let A be a type expression. We define $h(A)$, the *height* of A, and $r(A)$, the *rank* of A, as follows:

$$\begin{array}{ll} h(P) = h(X) = 0 & r(P) = r(X) = r(\bullet A) = 0 \\ h(\bullet A) = h(A) + 1 & \multirow{2}{*}{$r(A \to B) = \begin{cases} 0 & (B \text{ is a } \top\text{-variant}) \\ \max(r(A),\, r(B)) + 1 & (\text{otherwise}) \end{cases}$} \\ h(A \to B) = \max(h(A),\, h(B)) + 1 & \\ h(\mu X.A) = h(A) + 1 & r(\mu X.A) = r(A) + 1 \end{array}$$

Proposition 4.6. *If A is proper in X, then $r(A[B/X]) < r(\mu X.A)$ for every B.*

Proof. By straightforward induction on $h(A)$, and by cases on the form of A. □

Proposition 4.7. *1. If A and B are proper in X, then so is $A[B/Y]$.*
2. If A is proper in X, then so is $A[B/X]$ for every B.

Proof. By straightforward induction on $h(A)$, and by cases on the form of A. □

Definition 4.8. Let be A be a type expression. We define sets $ETV^{+}(A)$ and $ETV^{-}(A)$ of type variables as follows:

$$\begin{aligned}
ETV^{\pm}(P) &= \{\} \\
ETV^{+}(X) &= \{X\} \\
ETV^{-}(X) &= \{\} \\
ETV^{\pm}(\bullet A) &= ETV^{\pm}(A) \\
ETV^{\pm}(A \to B) &= \begin{cases} \{\} & (B \text{ is a } \top\text{-variant}) \\ ETV^{\mp}(A) \cup ETV^{\pm}(B) & (\text{otherwise}) \end{cases} \\
ETV^{\pm}(\mu X.A) &= \begin{cases} (ETV^{\pm}(A) \cup ETV^{\mp}(A)) - \{X\} & (X \in ETV^{-}(A)) \\ ETV^{\pm}(A) - \{X\} & (X \notin ETV^{-}(A)) \end{cases}
\end{aligned}$$

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 a $\top$-variant. Note that $ETV^{\pm}(A) \subset FTV(A)$.

Proposition 4.9. *1. If A is a $\top$-variant, then so is $A[B/X]$.*
2. If $A[B/X]$ is a $\top$-variant, then (a) A is also a $\top$-variant, or (b) $X \in ETV^{+}(A)$ and B is a $\top$-variant.

Proof. By straightforward induction on $h(A)$, and by cases on the form of A. □

Proposition 4.10. *1. If $X \in ETV^{\pm}(A)$, $X \neq Y$ and B is not a $\top$-variant, then $X \in ETV^{\pm}(A[B/Y])$.*
2. If $X \notin ETV^{\pm}(A)$ and $X \notin ETV^{+}(B) \cup ETV^{-}(B)$, then $X \notin ETV^{\pm}(A[B/Y])$.

Proof. By straightforward induction on $h(A)$, and by cases of the form of A. Use Proposition 4.9.2 for part 1 in case that $A = C \to D$ for some C and D. Note that $X \in ETV^{\pm}(C \to D)$ implies that D is not a $\top$-variant. □

Proposition 4.11. *1. If $X \in ETV^{+}(A)$ and $Y \in ETV^{\pm}(B)$, then $Y \in ETV^{\pm}(A[B/X])$.*
2. If $X \in ETV^{-}(A)$ and $Y \in ETV^{\pm}(B)$, then $Y \in ETV^{\mp}(A[B/X])$.
3. If $X \notin ETV^{+}(A)$ and $Y \notin ETV^{\pm}(A) \cup ETV^{\mp}(B)$, then $Y \notin ETV^{\pm}(A[B/X])$.
4. If $X \notin ETV^{-}(A)$ and $Y \notin ETV^{\pm}(A) \cup ETV^{\pm}(B)$, then $Y \notin ETV^{\pm}(A[B/X])$.

Proof. By simultaneous induction on $h(A)$. □

Definition 4.12. Let X be a type variable, and A a type expression. The *positive $\bullet$-depth $dp^{+}_{\bullet}(A, X)$*, the *negative $\bullet$-depth $dp^{-}_{\bullet}(A, X)$*, the *positive $\to$-depth $dp^{+}_{\to}(A, X)$* and the *negative $\to$-depth $dp^{-}_{\to}(A, X)$* of X in A are defined as follows:

$$\begin{aligned}
dp^{\pm}_{\bullet}(P, X) &= \infty \\
dp^{+}_{\bullet}(Y, X) &= \begin{cases} 0 & (X = Y) \\ \infty & (X \neq Y) \end{cases} \\
dp^{-}_{\bullet}(Y, X) &= \infty \\
dp^{\pm}_{\bullet}(\bullet A, X) &= dp^{\pm}_{\bullet}(A, X) + 1 \\
dp^{\pm}_{\bullet}(A \to B, X) &= \begin{cases} \infty & (B \text{ is a } \top\text{-variant}) \\ \min(dp^{\mp}_{\bullet}(A, X),\, dp^{\pm}_{\bullet}(B, X)) & (\text{otherwise}) \end{cases} \\
dp^{\pm}_{\bullet}(\mu Y.A, X) &= \min(dp^{\pm}_{\bullet}(A, X),\, dp^{-}_{\bullet}(A, Y) + dp^{\mp}_{\bullet}(A, X)) \qquad (X \neq Y)
\end{aligned}$$

$$\begin{aligned}
dp^{\pm}_{\to}(P, X) &= \infty \\
dp^{+}_{\to}(Y, X) &= \begin{cases} 0 & (X = Y) \\ \infty & (X \neq Y) \end{cases}
\end{aligned}$$

$$
\begin{aligned}
dp^{-}_{\to}(Y,\, X) &= \infty \\
dp^{\pm}_{\to}(\bullet A,\, X) &= dp^{\pm}_{\to}(A,\, X) \\
dp^{\pm}_{\to}(A \to B,\, X) &= \begin{cases} \infty & (B \text{ is a } \top\text{-variant}) \\ \min(dp^{\mp}_{\to}(A,\, X),\, dp^{\pm}_{\to}(B,\, X)) + 1 & (\text{otherwise}) \end{cases} \\
dp^{\pm}_{\to}(\mu Y.A,\, X) &= \min(dp^{\pm}_{\to}(A,\, X),\, dp^{-}_{\to}(A,\, Y) + dp^{\mp}_{\to}(A,\, X)) \qquad (X \neq Y)
\end{aligned}
$$

Note that $dp^{\pm}_{\bullet}(A,\, X)$, $dp^{\pm}_{\to}(A,\, X) \in \{0,\, 1,\, 2,\, \ldots,\, \infty\}$. We can easily check that α-conversions do not affect the definition of $dp^{\pm}_{\bullet}(A,\, X)$ or $dp^{\pm}_{\to}(A,\, X)$. We also define the *$\bullet$-depth* $dp_{\bullet}(A,\, X)$ and *$\to$-depth* $dp_{\to}(A,\, X)$ of X in A as follows:

$$
\begin{aligned}
dp_{\bullet}(A,\, X) &= \min(dp^{+}_{\bullet}(A,\, X),\, dp^{-}_{\bullet}(A,\, X)) \\
dp_{\to}(A,\, X) &= \min(dp^{+}_{\to}(A,\, X),\, dp^{-}_{\to}(A,\, X))
\end{aligned}
$$

Proposition 4.13. *Let dp be $dp_{\bullet}$ or $dp_{\to}$.*

1. $dp^{\pm}(A,\, X) < \infty$ *if and only if* $X \in ETV^{\pm}(A)$.
2. $dp^{\pm}(A[B/X],\, X) = \min(dp^{+}(A,\, X) + dp^{\pm}(B,\, X),\, dp^{-}(A,\, X) + dp^{\mp}(B,\, X))$.
3. *If* $X \neq Y$, *then* $dp^{\pm}(A[B/X],\, Y) = \min(dp^{\pm}(A,\, Y),\, dp^{+}(A,\, X) + dp^{\pm}(B,\, Y),\, dp^{-}(A,\, X) + dp^{\mp}(B,\, Y))$.

Proof. By induction on $h(A)$, and by cases of the form of A. □

Proposition 4.14. *A is proper in X if and only if $dp_{\bullet}(A,\, X) > 0$.*

Proof. By straightforward induction on $h(A)$, and by cases of the form of A. □

Proposition 4.15. 1. *If* $dp^{+}_{\to}(A, X) = 0$, *then* $ETV^{+}(A) = \{X\}$ *and* $ETV^{-}(A) = \{\}$.
2. $dp^{-}_{\to}(A, X) > 0$.

Proof. By straightforward simultaneous induction on $h(A)$, and by cases of the form of A. □

5 Equality of types

Definition 5.1 **($\simeq$).** The equivalence relation $\simeq$ over **TExp** 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 $A \to B \simeq C \to D$.
($\simeq$-$\to\top$)	$A \to \top \simeq B \to \top$.
($\simeq$-fix)	$\mu X.A \simeq A[\mu X.A/X]$.
($\simeq$-uniq)	If $A \simeq C[A/X]$ and C is proper in X, then $A \simeq \mu X.C$.

Note that except for ($\simeq$-$\bullet$), ($\simeq$-$\to\top$) and ($\simeq$-uniq), these conditions are the same as $\simeq_{\lambda\mu}$. Two type expressions of $\lambda\bullet\mu$ are equivalent modulo $\simeq$, if their (possibly infinite) type expression obtained by indefinite unfolding recursive types occurring in them are identical modulo the rule ($\simeq$-$\to\top$).

Proposition 5.2. *Let n be a non-negative integer, X_1, X_2, ..., X_n type variables, and A, B_1, B_2, ..., B_n, C_1, C_2, ..., C_n type expressions of $\lambda\mu$. If $B_i \simeq C_i$ for every i ($i = 1,\, 2,\, \ldots,\, n$), then $A[B_1/X_1,\, B_2/X_2,\, \ldots,\, B_n/X_n] \simeq A[B_1/X_1,\, B_2/X_2,\, \ldots,\, B_n/X_n]$.*

Proof. Similar to the proof of Proposition 3.4. Use Proposition 4.7.2 for the case that $A = \mu Y.A'$ for some Y and A'. □

Proposition 5.3. *Let n be a non-negative integer, X_1, X_2, ..., X_n type variables, and A, B, C_1, C_2, ..., C_n, D_1, D_2, ..., D_n, type expressions of $\lambda\mu$. If $A \simeq B$ and $C_i \simeq D_i$ for every i ($i = 1,\, 2,\, \ldots,\, n$), then $A[C_1/X_1,\, C_2/X_2,\, \ldots,\, C_n/X_n] \simeq B[D_1/X_1,\, D_2/X_2,\, \ldots,\, D_n/X_n]$.*

Proof. Similar to the proof of Proposition 3.5. Use Proposition 4.7.2 for the case that the last rule is ($\simeq$-fix) or ($\simeq$-uniq) $\square$

In the sequel, we use Proposition 5.3 in proofs without notice.

Proposition 5.4. *If $A \simeq B$, then $dp^{\pm}_{\bullet}(A, X) = dp^{\pm}_{\bullet}(B, X)$ and $dp^{\pm}_{\to}(A, X) = dp^{\pm}_{\to}(B, X)$.*

Proof. By induction on the derivation of $A \simeq B$. The only interesting case is when the last rule is ($\simeq$-uniq). Suppose that $A \simeq B$. There exist some Y and C such that $B = \mu Y.C$, $A \simeq C[A/Y]$ and C is proper in Y. By the induction hypothesis,

$$dp^{\pm}_{\bullet}(A, Z) = dp^{\pm}_{\bullet}(C[A/Y], Z) \text{ and } dp^{\pm}_{\to}(A, Z) = dp^{\pm}_{\to}(C[A/Y], Z) \text{ for every } Z. \tag{1}$$

Therefore, $dp^{\pm}_{\bullet}(A, Y) = dp^{\pm}_{\bullet}(C[A/Y], Y) = \min(dp^{+}_{\bullet}(C, Y) + dp^{\pm}_{\bullet}(A, Y), dp^{-}_{\bullet}(C, Y) + dp^{\mp}_{\bullet}(A, Y))$ by Proposition 4.13.2. Since $dp^{\pm}_{\bullet}(C, Y) > 0$ by Proposition 4.14, we get $dp^{\pm}_{\bullet}(A, Y) = \infty$; and therefore, also $dp^{\pm}_{\to}(A, Y) = \infty$ by Propositions 4.13.1. If $X = Y$, then $dp^{\pm}_{\bullet}(\mu Y.C, X) = \infty = dp^{\pm}_{\bullet}(A, X)$ and $dp^{\pm}_{\to}(\mu Y. C, X) = \infty = dp^{\pm}_{\to}(A, X)$ by Propositions 4.13.1. Otherwise, i.e., if $X \neq Y$, then

$$\begin{aligned}
dp^{\pm}_{\bullet}(A, X) &= dp^{\pm}_{\bullet}(C[A/Y], X) && \text{(by (1))}\\
&= \min(dp^{\pm}_{\bullet}(C, X), dp^{+}_{\bullet}(C, Y) + dp^{\pm}_{\bullet}(A, X), dp^{-}_{\bullet}(C, Y) + dp^{\mp}_{\bullet}(A, X)))\\
& && \text{(by Proposition 4.13.3)}\\
&= \min(dp^{\pm}_{\bullet}(C, X), dp^{-}_{\bullet}(C, Y) + dp^{\mp}_{\bullet}(A, X)) && \text{(since } dp^{+}_{\bullet}(C, Y) > 0)\\
&= \min(dp^{\pm}_{\bullet}(C, X), dp^{-}_{\bullet}(C, Y) + \min(dp^{\mp}_{\bullet}(C, X), dp^{-}_{\bullet}(C, Y) + dp^{\pm}_{\bullet}(A, X)))\\
&= \min(dp^{\pm}_{\bullet}(C, X), dp^{-}_{\bullet}(C, Y) + dp^{\mp}_{\bullet}(C, X), dp^{-}_{\bullet}(C, Y) + dp^{-}_{\bullet}(C, Y) + dp^{\pm}_{\bullet}(A, X))\\
&= \min(dp^{\pm}_{\bullet}(C, X), dp^{-}_{\bullet}(C, Y) + dp^{\mp}_{\bullet}(C, X)) && \text{(since } dp^{-}_{\bullet}(C, Y) > 0)\\
&= dp^{\pm}_{\bullet}(\mu Y.C, X) && \text{(by Definition 4.12)}
\end{aligned}$$

Similarly, we get $dp^{\pm}_{\to}(A, X) = \min(dp^{\pm}_{\to}(C, X), dp^{+}_{\to}(C, Y) + dp^{\pm}_{\to}(A, X), dp^{-}_{\to}(C, Y) + dp^{\mp}_{\to}(A, X))$ in this case. If $dp^{\pm}_{\to}(C, Y) > 0$, then $dp^{\pm}_{\to}(A, X) = dp^{\pm}_{\to}(\mu Y.C, X)$ as in the case of $dp^{\pm}_{\bullet}$. Otherwise, i.e., if $dp^{+}_{\to}(C, Y) = 0$, then $X \notin FTV(\mu Y.C)$ by Propositions 4.15; and therefore $X \notin FTV(A)$ by Propositions 4.13.1, because we already have $dp^{\pm}_{\bullet}(A, X) = dp^{\pm}_{\bullet}(\mu Y.C, X)$. We now get $dp^{\pm}_{\to}(A, X) = \infty = dp^{\pm}_{\to}(\mu Y.C, X)$ by Propositions 4.13.1 again. $\square$

Proposition 5.5. *Suppose that $A \simeq B$.*

1. *$ETV^{\pm}(A) = ETV^{\pm}(B)$.*
2. *A is proper in X if and only if so is B.*

Proof. Obvious from Propositions 4.13.1, 4.14 and 5.4. $\square$

Definition 5.6. Let n be a non-negative integer, and $X_1, X_2, \ldots, X_n$ be type variables. Let A and B be type expressions. We define a type expression $A[B/X_1, X_2 \ldots X_n]$ by $A[B/X'_1, B/X'_2 \ldots B/X'_m]$, where $X'_1, X'_2 \ldots X'_m$ are distinct type variables such that $\{X'_1, X'_2 \ldots X'_m\} = \{X_1, X_2 \ldots X_n\}$. We use $A[B/\boldsymbol{X}]$ to denote a type expression of the form $A[B/X_1, X_2 \ldots X_n]$.

Proposition 5.7. $\mu\boldsymbol{X}A \simeq A[\mu\boldsymbol{X}A/\boldsymbol{X}]$

Proof. By induction on the length of $\boldsymbol{X}$. Let $\boldsymbol{X} = X_1, \boldsymbol{X'}$.

$$\begin{aligned}
\mu\boldsymbol{X}A &\simeq (\mu\boldsymbol{X'}.A)[\mu\boldsymbol{X}A/X_1] && \text{(by ($\simeq$-fix))}\\
&\simeq A[\mu\boldsymbol{X'}.A/\boldsymbol{X'}][\mu\boldsymbol{X}A/X_1] && \text{(by ind. hyp.)}\\
&= A[\mu\boldsymbol{X}A/X_1][(\mu\boldsymbol{X'}.A)[\mu\boldsymbol{X}A/X_1]/\boldsymbol{X'}]\\
&\simeq A[\mu\boldsymbol{X}A/X_1][\mu\boldsymbol{X}A/\boldsymbol{X'}] && \text{(by ($\simeq$-fix))}\\
&= A[\mu\boldsymbol{X}A/X_1, \boldsymbol{X'}]\\
&= A[\mu\boldsymbol{X}A/\boldsymbol{X}]
\end{aligned}$$

$\square$

Proposition 5.8. *1. If $A \simeq B$ and A or B is proper in X, then $\mu X.A \simeq \mu X.B$.*
2. If $A \simeq C[A/X]$, $B \simeq C[B/X]$, and C is proper in X, then $A \simeq B$.
3. $\mu X.A[X/Y] \simeq \mu X.A[A[X/Y]/Y]$.
4. $\mu X.A[X/Y] \simeq \mu X.\mu Y.A$.

Proof. For the part 1, we get $\mu X.A \simeq A[\mu X.A/X] \simeq B[\mu X.A/X]$ by ($\simeq$-fix) and Proposition 5.3. Note that if A is proper in X, then so is B by Proposition 5.5 since $A \simeq B$. Therefore, $\mu X.A \simeq \mu X.B$ by ($\simeq$-uniq). The part 2 is obvious by ($\simeq$-uniq), ($\simeq$-symm) and ($\simeq$-trans). For the part 3, let $C = A[A[X/Y]/Y]$ and show that $\mu X. A[X/Y] \simeq C[\mu X.A[X/Y]/X]$ and $\mu X.A[A[X/Y]/Y] \simeq C[\mu X.A[A[X/Y]/Y]/X]$. Note that $A[A[X/Y]/Y]$ is proper in X by Proposition 4.7. For the part 4, let $C = A[X/Y]$ and show that $\mu X.A[X/Y] \simeq C[\mu X.A[X/Y]/X]$ and $\mu X.\mu Y.A \simeq C[\mu X.\mu Y.A/X]$. Use Proposition 5.7.

Proposition 5.9. *1. $\top \simeq \bullet\top$.*
2. $\top \simeq \mu X.\bullet^n X$ for every $n \geq 1$.

Proof. Obvious from ($\simeq$-fix) and ($\simeq$-uniq). ◻

Proposition 5.10. *$\mu \boldsymbol{Z} A \to B \simeq C$ if and only if $C = \mu \boldsymbol{Z'}.A' \to B'$ for some $\boldsymbol{Z'}$, A' and B' such that:*

$$B[\mu \boldsymbol{Z} A \to B/\boldsymbol{Z}] \simeq B'[\mu \boldsymbol{Z'}.A' \to B'/\boldsymbol{Z'}], \text{ and} \tag{2}$$
$$A[\mu \boldsymbol{Z} A \to B/\boldsymbol{Z}] \simeq A'[\mu \boldsymbol{Z'}.A' \to B'/\boldsymbol{Z'}] \text{ or } B[\mu \boldsymbol{Z} A \to B/\boldsymbol{Z}] \simeq \top \tag{3}$$

Proof. First, we show the "if" part. Suppose that there exist some $\boldsymbol{Z'}$, A' and B' such that $C = \mu \boldsymbol{Z'}.A' \to B'$, and both (2) and (3) hold.

$$\begin{aligned}
\mu \boldsymbol{Z} A \to B &\simeq (A \to B)[\mu \boldsymbol{Z} A \to B/\boldsymbol{Z}] && \text{(by Proposition 5.7)}\\
&= A[\mu \boldsymbol{Z} A \to B/\boldsymbol{Z}] \to B[\mu \boldsymbol{Z'}.A \to B/\boldsymbol{Z'}] \\
&\simeq A'[\mu \boldsymbol{Z'}.A' \to B'/\boldsymbol{Z'}] \to B'[\mu \boldsymbol{Z'}.A' \to B'/\boldsymbol{Z'}] && \text{(by (2), (3) and ($\simeq$-$\to$))}\\
&= (A' \to B')[\mu \boldsymbol{Z'}.A' \to B'/\boldsymbol{Z'}] \\
&\simeq \mu \boldsymbol{Z'}.A' \to B' && \text{(by Proposition 5.7)}\\
&= C
\end{aligned}$$

As for the "only if" part, we prove a more general statement: if $\mu \boldsymbol{Z} A \to B \simeq C$ or $C \simeq \mu \boldsymbol{Z} A \to B$, then $C = \mu \boldsymbol{Z'}.A' \to B'$ for some $\boldsymbol{Z'}$, A' and B' such that (2) and (3) hold. The proof proceeds by induction on the derivation of $\mu \boldsymbol{Z} A \to B \simeq C$ or $C \simeq \mu \boldsymbol{Z} A \to B$, and by cases of the last rule used in the derivation.

Case: ($\simeq$-μ). Let $\boldsymbol{Z} = X, \boldsymbol{Y}$. Since $\mu X.\mu \boldsymbol{Y}.A \to B \simeq C$ or $C \simeq \mu X.\mu \boldsymbol{Y}.A \to B$ is derivable, so is $\mu \boldsymbol{Y}.A \to B \simeq C'$ or $C' \simeq \mu \boldsymbol{Y}.A \to B$ for some C' such that $C = \mu X.C'$. By the induction hypothesis, $C' = \mu \boldsymbol{Y'}.A' \to B'$ for some A' and B' such that:

$$B[\mu \boldsymbol{Y}.A \to B/\boldsymbol{Y}] \simeq B'[\mu \boldsymbol{Y'}.A' \to B'/\boldsymbol{Y'}], \text{ and} \tag{4}$$
$$A[\mu \boldsymbol{Y}.A \to B/\boldsymbol{Y}] \simeq A'[\mu \boldsymbol{Y'}.A' \to B'/\boldsymbol{Y'}] \text{ or } B[\mu \boldsymbol{Y}.A \to B/\boldsymbol{Y}] \simeq \top \tag{5}$$

We can assume that $X \notin \{\boldsymbol{Y}, \boldsymbol{Y'}\}$ without loss of generality.

$$\begin{aligned}
B[\mu X.\mu \boldsymbol{Y}.A \to B/X, \boldsymbol{Y}] &= B[\mu X.\mu \boldsymbol{Y}.A \to B/\boldsymbol{Y}][\mu X.\mu \boldsymbol{Y}.A \to B/X] \\
&\simeq B[(\mu \boldsymbol{Y}.A \to B)[\mu X.\mu \boldsymbol{Y}.A \to B/X]/\boldsymbol{Y}][\mu X.\mu \boldsymbol{Y}.A \to B/X] && \text{(by ($\simeq$-fix))}\\
&= B[\mu \boldsymbol{Y}.A \to B/\boldsymbol{Y}][\mu X.\mu \boldsymbol{Y}.A \to B/X] \\
&\simeq B'[\mu \boldsymbol{Y'}.A' \to B'/\boldsymbol{Y'}][\mu X.\mu \boldsymbol{Y}.A \to B/X] && \text{(by (4))}\\
&\simeq B'[\mu \boldsymbol{Y'}.A' \to B'/\boldsymbol{Y'}][\mu X.\mu \boldsymbol{Y'}.A' \to B'/X] && \text{(since $\mu \boldsymbol{Y}.A \to B \simeq C'$)}\\
&= B'[(\mu \boldsymbol{Y'}.A' \to B')[\mu X.\mu \boldsymbol{Y'}.A' \to B'/X]/\boldsymbol{Y'}][\mu X.\mu \boldsymbol{Y'}.A' \to B'/X] \\
&\simeq B'[\mu X.\mu \boldsymbol{Y'}.A' \to B'/\boldsymbol{Y'}][\mu X.\mu \boldsymbol{Y'}.A' \to B'/X] && \text{(by ($\simeq$-fix))}\\
&= B'[\mu X.\mu \boldsymbol{Y'}.A' \to B'/X, \boldsymbol{Y'}]
\end{aligned}$$

Similarly, we get $A[\mu X.\mu \boldsymbol{Y}.A \to B/X, \boldsymbol{Y}] = A'[\mu X.\mu \boldsymbol{Y'}.A' \to B'/X, \boldsymbol{Y'}]$ or $B[\mu X.\mu \boldsymbol{Y}.A \to B/X, \boldsymbol{Y}] \simeq \top$ from (5).

Case: ($\simeq$*-fix*). For some X, $\boldsymbol{Y}$, D and E, the derivation ends with $\mu X.\mu\boldsymbol{Y}.D \to E \simeq (\mu\boldsymbol{Y}.D \to E)[\mu X.\mu\boldsymbol{Y}.D \to E/X]$. Since we can assume that $X \notin \{\boldsymbol{Y}\}$ without loss of generality, we get $(\mu\boldsymbol{Y}.D \to E)[\mu X.\mu\boldsymbol{Y}.D \to E/X] = \mu\boldsymbol{Y}.(D \to E)[\mu X.\mu\boldsymbol{Y}.D \to E/X]$.

$$\begin{aligned}
E[\mu X.\mu\boldsymbol{Y}.D \to E/X,\, \boldsymbol{Y}] &= E[\mu X.\mu\boldsymbol{Y}.D \to E/X][\mu X.\mu\boldsymbol{Y}.D \to E/\boldsymbol{Y}] \\
&\simeq E[\mu X.\mu\boldsymbol{Y}.D \to E/X][(\mu\boldsymbol{Y}.D \to E)[\mu X.\mu\boldsymbol{Y}.D \to E/X]/\boldsymbol{Y}] && \text{(by ($\simeq$-fix))} \\
&= E[\mu X.\mu\boldsymbol{Y}.D \to E/X][\mu\boldsymbol{Y}.(D \to E)[\mu X.\mu\boldsymbol{Y}.D \to E/X]/\boldsymbol{Y}]
\end{aligned}$$

Similarly, we get $D[\mu X.\mu\boldsymbol{Y}.D \to E/X,\, \boldsymbol{Y}] \simeq D[\mu X.\mu\boldsymbol{Y}.D \to E/X][\mu\boldsymbol{Y}.(D \to E)[\mu X.\mu\boldsymbol{Y}.D \to E/X]/\boldsymbol{Y}]$ or $E[\mu X.\mu\boldsymbol{Y}.D \to E/X,\, \boldsymbol{Y}] \simeq \top$.

Case: ($\simeq$*-uniq*). If $\mu\boldsymbol{Z}A \to B \simeq C$, there exists some C' such that $C = \mu X.C'$, $\mu\boldsymbol{Z}A \to B \simeq C'[\mu\boldsymbol{Z}A \to B/X]$, and C' is proper in X. In this case, by the induction hypothesis, we get $C' = \mu\boldsymbol{Y}.A' \to B'$ for some $\boldsymbol{Y}$, A' and B'. On the other hand, if $C \simeq \mu\boldsymbol{Z}A \to B$, then we have $C \simeq (\mu\boldsymbol{Y}.A \to B)[C/X]$ for some $\boldsymbol{Y}$ such that $\boldsymbol{Z} = X, \boldsymbol{Y}$. In this case, $C = \mu\boldsymbol{Y'}.A' \to B'$ for some $\boldsymbol{Y'}$ A' and B' by the induction hypothesis. Therefore, in either case, the derivation ends with:

$$\frac{\mu\boldsymbol{Z}A \to B \simeq (\mu\boldsymbol{Y}.A' \to B')[\mu\boldsymbol{Z}A \to B/X]}{\mu\boldsymbol{Z}A \to B \simeq \mu X.\mu\boldsymbol{Y}.A' \to B'}\ (\simeq\text{-uniq})$$

for some $\boldsymbol{Y}$, D and E such that $X \notin \{\boldsymbol{Y}\}$. Let $\boldsymbol{Z'} = X, \boldsymbol{Y}$. Then it suffices to show that (2) and (3) hold. We can assume that $X \notin \{\boldsymbol{Z}\}$ without loss of generality. By the induction hypothesis,

$$B[\mu\boldsymbol{Z}A \to B/\boldsymbol{Z}] \simeq B'[\mu\boldsymbol{Z}A \to B/X][(\mu\boldsymbol{Y}.A' \to B')[\mu\boldsymbol{Z}A \to B/X]/\boldsymbol{Y}], \text{ and} \tag{6}$$
$$A[\mu\boldsymbol{Z}A \to B/\boldsymbol{Z}] \simeq A'[\mu\boldsymbol{Z}A \to B/X][(\mu\boldsymbol{Y}.A' \to B')[\mu\boldsymbol{Z}A \to B/X]/\boldsymbol{Y}] \text{ or } B[\mu\boldsymbol{Z}A \to B/\boldsymbol{Z}] \simeq \top \tag{7}$$

Therefore,

$$\begin{aligned}
B[\mu\boldsymbol{Z}A \to B/\boldsymbol{Z}] &\simeq B'[\mu\boldsymbol{Z}A \to B/X][(\mu\boldsymbol{Y}.A' \to B')[\mu\boldsymbol{Z}A \to B/X]/\boldsymbol{Y}] && \text{(by (6))} \\
&= B'[\mu\boldsymbol{Y}.A' \to B'/\boldsymbol{Y}][\mu\boldsymbol{Z}A \to B/X] \\
&\simeq B'[\mu\boldsymbol{Y}.A' \to B'/\boldsymbol{Y}][\mu X.\mu\boldsymbol{Y}.A' \to B'/X] \\
&= B'[(\mu\boldsymbol{Y}.A' \to B')[\mu X.\mu\boldsymbol{Y}.A' \to B'/X]/\boldsymbol{Y}][\mu X.\mu\boldsymbol{Y}.A' \to B'/X] \\
&\simeq B'[\mu X.\mu\boldsymbol{Y}.A' \to B'/\boldsymbol{Y}][\mu X.\mu\boldsymbol{Y}.A' \to B'/X] && \text{(by ($\simeq$-fix))} \\
&= B'[\mu X.\mu\boldsymbol{Y}.A' \to B'/X,\, \boldsymbol{Y}] \\
&= B'[\mu X.\mu\boldsymbol{Y}.A' \to B'/\boldsymbol{Z'}]
\end{aligned}$$

Similarly, we also get $A[\mu\boldsymbol{Z}A \to B/\boldsymbol{Z}] \simeq A'[\mu X.\mu\boldsymbol{Y}.A' \to B'/\boldsymbol{Z'}]$ or $B[\mu\boldsymbol{Z}A \to B/\boldsymbol{Z}] \simeq \top$. Other cases are rather straightforward. ☐

Proposition 5.11. *Let P be a type constant. $\mu\boldsymbol{Z}P \simeq C$ if and only if $C = \mu\boldsymbol{Z'}.P$ for some $\boldsymbol{Z'}$.*

Proof. Similar to the proof of Proposition 5.10. ☐

Proposition 5.12. *Let $X \notin \{\boldsymbol{Z}\}$. $\mu\boldsymbol{Z}X \simeq C$ if and only if $C = \mu\boldsymbol{Z'}.X$ for some $\boldsymbol{Z'}$ such that $X \notin \{\boldsymbol{Z'}\}$.*

Proof. Similar to the proof of Proposition 5.10. ☐

Proposition 5.13. *$\mu\boldsymbol{Z}\bullet A \simeq C$ if and only if $C = \mu\boldsymbol{Z'}.\bullet A'$ for some $\boldsymbol{Z'}$ and A' such that $A[\mu\boldsymbol{Z}\bullet A/\boldsymbol{Z}] \simeq A'[\mu\boldsymbol{Z'}.\bullet A'/\boldsymbol{Z'}]$.*

Proof. Similar to the proof of Proposition 5.10. ☐

Proposition 5.14. *1. $\bullet A \simeq \bullet B$ if and only if $A \simeq B$.*
2. $A \to B \simeq C \to D$ if and only if (a) $B \simeq D \simeq \top$, or (b) $A \simeq C$ and $B \simeq D$.

Proof. "If" part is obvious from ($\simeq$-$\bullet$) and ($\simeq$-$\to$). We get "only if" part from Propositions 5.10 and 5.13. ☐

Proposition 5.15. *1. $\top \not\simeq \bullet^n P,\ \bullet^n X,\ \bullet^n(A \to B)$ for any $n \geq 0$.*
2. $\bullet A \not\simeq P,\ X,\ B \to C$

3. $P \not\simeq X,\ A \to B$
4. $X \not\simeq A \to B$

Proof. Obvious from Propositions 5.10, 5.13, 5.11 and 5.12. □

Proposition 5.16. *If* $\bullet A \simeq \top$*, then* $A \simeq \top$.

Proof. Obvious from Proposition 5.14.1 because $\top \simeq \bullet\top$ by Proposition 5.9.1. □

Proposition 5.17. *If* $\bullet^n A \simeq A$*, then (a)* $A \simeq \top$ *or (b)* $n = 0$.

Proof. Suppose that $\bullet^n A \simeq A$ and $n > 0$. Since $\bullet^n X$ is proper in X, we get $A \simeq \mu X.\bullet^n X$ by ($\simeq$-uniq). Therefore, $A \simeq \top$ by Proposition 5.9.2. □

Proposition 5.18. *Suppose that* $A \not\simeq X$ *and* $m \geq n$.

1. *If* $A[B/X] \simeq \bullet^n P$ *and* $B \simeq \bullet^m P$*, then* $A \simeq \bullet^n P$.
2. *If* $A[B/X] \simeq \bullet^n Y$ *and* $B \simeq \bullet^m Y$*, then* $A \simeq \bullet^n Y$.
3. *If* $A[B/X] \simeq \bullet^n(C \to D)$ *and* $B \simeq \bullet^m(C \to D)$*, then* $A \simeq \bullet^n(C' \to D')$ *for some* C' *and* D' *such that (a)* $D'[B/X] \simeq D \simeq \top$*, or (b)* $C'[B/X] \simeq C$ *and* $D'[B/X] \simeq D$.

Proof. By induction on the lexicographic ordering of $<n,\ r(A)>$, and by cases of the form of A. Use Propositions 5.10, 5.11, 5.12 and 5.13. The induction hypothesis is used only if A is in the form of $\bullet A'$ or $\mu Z.A'$. Suppose that $A[B/X] \simeq \bullet^n E$ and $B \simeq \bullet^m E$, where E is either P, Y or $C \to D$.

Case: A is a type constant. Since $A[B/X] = A \simeq \bullet^n E$, we get $n = 0$ and $E = P$ by Proposition 5.15; and therefore, $A \simeq P$.

Case: A is a type variable. Since $A \not\simeq X$, we get $n = 0$ and $E = Y$ from $A[B/X] \simeq \bullet^n E$ by Proposition 5.15; and therefore, $A \simeq Y$.

Case: $A = \bullet A'$ for some A'. In this case, $n > 0$ and $A'[B/X] \simeq \bullet^{n-1}E$ by Propositions 5.14.1 and 5.15. If $A' \simeq X$, then $\bullet^n E \simeq A[B/X] \simeq \bullet B \simeq \bullet^{m+1}E$; however, this is impossible by Propositions 5.14.1 and 5.15 since $m{+}1 > n$. Therefore, $A' \not\simeq X$. If $E = P$ or $E = X$, then we get $A' \simeq \bullet^{n-1}E$, i.e., $A \simeq \bullet^n E$ by the induction hypothesis. Similarly, if $E = C \to D$, then we get $A' \simeq \bullet^{n-1}(C' \to D')$ i.e., $A \simeq \bullet^n(C' \to D')$, for some C' and D' such that (a) $D'[B/X] \simeq D \simeq \top$, or (b) $C'[B/X] \simeq C$ and $D'[B/X] \simeq D$.

Case: $A = A_1 \to A_2$ for some A_1 and A_2. In this case, $n = 0$ and $E = C \to D$ by Proposition 5.15. Therefore, taking C' and D' as $C' = A_1$ and $D' = A_2$, we get $A \simeq \bullet^n(C' \to D')$; and since $C \to D = E = \bullet^n E \simeq A[B/X] = C'[B/X] \to D'[B/X]$, we also get (a) $D'[B/X] \simeq D \simeq \top$, or (b) $C'[B/X] \simeq C$ and $D'[B/X] \simeq D$, by Propositions 5.14.2.

Case: $A = \mu Z.A'$ for some Z and A'. We can assume $Z \notin FTV(B) \cup \{X\}$. Since $A \simeq A'[\mu Z.A'/Z]$, we get $A'[\mu Z.A'] \not\simeq X$ and $A'[\mu Z.A'/Z][B/X] \simeq (\mu Z.A')[B/X] = A[B/X] \simeq \bullet^n E$. Note also that $r(A'[\mu Z.A'/Z]) < r(A)$. Therefore, if $E = P$ or $E = X$, then we get $A \simeq A'[\mu Z.A'/Z] \simeq \bullet^n E$ by the induction hypothesis. Similarly, if $E = C{\to}D$, then we get $A \simeq A'[\mu Z.A'/Z] \simeq \bullet^n(C'{\to}D')$ for some C' and D' such that (a) $D'[B/X] \simeq D \simeq \top$, or (b) $C'[B/X] \simeq C$ and $D'[B/X] \simeq D$. □

Proposition 5.19. *Let* $X_1,\ X_2,\ \ldots,\ X_m$ *be distinct type variables. If* $A[B_1/X_1,\ B_2/X_2,\ \ldots,\ B_m/X_m] \simeq \top$ *and* $X_i \in FTV(A)$ $(1 \leq i \leq m)$*, then* $A \simeq \bullet^n X_i$ *for some* $n \geq 0$.

Proof. By induction on $h(A)$ and by cases of the form of A. Suppose that $A[B_1/X_1,\ B_2/X_2,\ \ldots,\ B_m/X_m] \simeq \top$ and $X_i \in FTV(A)$ $(1 \leq i \leq m)$.

Case: $A = P$ for some P, or $A = C \to D$ for some C and D. This case is impossible because $A[B/X] \not\simeq \top$ for any B by Proposition 5.15.1.

Case: $A = Y$ for some Y. Trivial because we get $Y = X_i$ from $X_i \in FTV(A)$.

Case: $A = \bullet A'$ for some A'. In this case, we can get $X_i \in FTV(A')$, and $A'[B_1/X_1, B_2/X_2, \ldots, B_m/X_m] \simeq \top$ from $\bullet A'[B_1/X_1, B_2/X_2, \ldots, B_m/X_m] = \top$ by Proposition 5.16. Since $h(A') < h(A)$, by the induction hypothesis, we get $A' \simeq \bullet^n X_i$ for some n; and therefore, $A = \bullet A' \simeq \bullet^{n+1} X_i$.

Case: $A = \mu Y.A'$ for some Y and A'. We can assume that $Y \not\equiv X_j$ for any j ($1 \le j \le m$). From the assumption, we get $X_i \in FTV(A')$ and

$$\begin{aligned}\top &\simeq A[B_1/X_1, B_2/X_2, \ldots, B_m/X_m]\\ &\simeq A'[\mu Y.A'/Y][B_1/X_1, B_2/X_2, \ldots, B_m/X_m]\\ &\simeq A'[(\mu Y.A')[B_1/X_1, B_2/X_2, \ldots, B_m/X_m]/Y, B_1/X_1, B_2/X_2, \ldots, B_m/X_m].\end{aligned}$$

Since $h(A') < h(A)$, by the induction hypothesis, we get $A' \simeq \bullet^n X_i$; and hence, $A \simeq \mu Y.\bullet^n X_i \simeq \bullet^n X_i$. □

Proposition 5.20. *A type expression A is a $\top$-variant if and only if $A \simeq \top$.*

Proof. The proof for the "only if" proceeds by straightforward induction on $h(A)$. Note that $\mu X.\bullet^{n+1} X \simeq \top$ by ($\simeq$-uniq) since $(\bullet^{n+1} X)[\top/X] \simeq \top$. For "if" part, it suffices to show the following:

1. If $A \simeq \top$, then A is a $\top$-variant.
2. If $A \simeq \bullet^n X$, then A is a $(\bullet^n X)$-variant,

where a type expression A is a $(\bullet^n X)$-variant if and only if $A = \bullet^{m_0} \mu Z_1.\bullet^{m_1} \mu Z_2.\bullet^{m_2} \ldots \mu Z_k.\bullet^{m_k} X$ for some k, m_0, m_1, m_2, …, m_k, Z_1, Z_2, …, Z_k such that $m_0+m_1+m_2+\ldots+m_k = n$ and $Z_i \not\equiv X$ for any i ($1 \le i \le k$). The proof proceeds by simultaneous induction on $h(A)$ and by cases of the form of A.

Case: $A = P$ for some P, or $A = C \to D$ for some C and D. This case is impossible because neither $A \simeq \top$ nor $A \simeq \bullet^n X$ holds by Proposition 5.15.

Case: $A = Y$ for some Y. In this case, we get $A \not\simeq \top$ by Proposition 5.15. If $A \simeq \bullet^n X$, then we get $Y = X$ and $n = 0$ also by Proposition 5.15, that is, A is a $(\bullet^n X)$-variant.

Case: $A = \bullet A'$ for some A'. If $A \simeq \top$, then $A' \simeq \top$ by Proposition 5.16; and therefore, A' is a $\top$-variant by the induction hypothesis. Hence, so is A. On the other hand, if $A \simeq \bullet^n X$, then $n > 0$ and $A' \simeq \bullet^{n-1} X$ by Propositions 5.13 and 5.14.1. Therefore, A' is a $(\bullet^{n-1} X)$-variant by the induction hypothesis, that is, A is a $(\bullet^n X)$-variant.

Case: $A = \mu Y.A'$ for some Y and A'. To show the first part, suppose that $A \simeq \top$. If $Y \notin FTV(A')$, then we get $A' \simeq \top$ since $A \simeq A'[\mu Y.A'/Y] = A'$ in this case. Therefore, A' is a $\top$-variant by the induction hypothesis; and so is A. On the other hand, if $Y \in FTV(A')$, then since $A'[\mu Y.A'/Y] \simeq A \simeq \top$, we get $A' \simeq \bullet^n Y$ for some n by Proposition 5.19; and hence, A' is a $(\bullet^n Y)$-variant by the induction hypothesis. A is thus a $\top$-variant since we get $n > 0$ from the fact that A' is proper in Y.

For the second part, suppose that $A \simeq \bullet^n X$, that is, $A'[A/Y] \simeq \bullet^n X$. We can assume that $Y \not\equiv X$. Since A' is proper in Y, $A' \not\simeq Y$. Hence, $A' \simeq \bullet^n X$ by Proposition 5.18.2; and therefore, A' is a $(\bullet^n X)$-variant by the induction hypothesis; and so is A since $Y \not\equiv X$. □

In the sequel, we use Proposition 5.20 in proofs without notice.

Definition 5.21 (Canonical types). We define a set **CTExp** of *canonical* type expressions as follows:

$$\mathbf{CTExp} \ ::= \ \top \ \mid \ \bullet^n\mathbf{TConst} \ \mid \ \bullet^n\mathbf{TVar} \ \mid \ \bullet^n(\mathbf{TExp} \to \mathbf{TExp}),$$

where n is an arbitrary non-negative integer.

Definition 5.22. We define A^c for each type expression A as follows:

$$\begin{aligned}P^c &= P\\ X^c &= X\\ (A \to B)^c &= A \to B\\ (\bullet A)^c &= \begin{cases}\top & (\text{if } A^c = \top)\\ \bullet A^c & (\text{otherwise})\end{cases}\\ (\mu X.A)^c &= \begin{cases}\top & (\text{if } A^c = \bullet^n X \text{ for some } n \ge 1)\\ A^c[\mu X.A/X] & (\text{otherwise})\end{cases}\end{aligned}$$

Proposition 5.23. *A^c is a canonical type expression such that $A^c \simeq A$.*

Proof. By induction on $h(A)$, and by cases of the form of A. Note that if a canonical type expression A does not have a form of $\bullet^n X$, then $A[B/X]$ is also canonical for any B. □

That is, Definition 5.22 can be regarded as an effective procedure for calculating a canonical type expression A^c such that $A^c \simeq A$ from a given type expression A of $\lambda\bullet\mu$.

Proposition 5.24. *If A is a canonical form, then $A^c = A$.*

Proof. By induction on $h(A)$, and by cases of the form of A. □

6 Subtyping induced by the modality

As mentioned in Introduction, our intended interpretation of the $\bullet$-modality introduces a subtyping relation into **TExp**. We now define the subtyping relation by a set of inference rules as in [3].

Definition 6.1. 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 $\{\, <X_i,\, Y_i> \mid i = 1,\, 2,\, \ldots,\, n\,\}$. We 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 6.2 ($\preceq$)**.** We define the derivability of *subtyping judgment* $\gamma \vdash A \preceq B$ by the following derivation rules:

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

$$\frac{}{\gamma \vdash A \preceq A'}\ (\preceq\text{-reflex})\quad (A \simeq A') \qquad \frac{\gamma_1 \vdash A \preceq B \quad \gamma_2 \vdash B \preceq C}{\gamma_1 \cup \gamma_2 \vdash A \preceq C}\ (\preceq\text{-trans})$$

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

$$\frac{}{\gamma \vdash A \preceq \bullet A}\ (\preceq\text{-approx}) \qquad \frac{}{\gamma \vdash A \to B \preceq \bullet A \to \bullet B}\ (\preceq\text{-}\to\bullet) \qquad \frac{}{\gamma \vdash \bullet A \to \bullet B \preceq \bullet(A \to B)}\ (\preceq\text{-}\bullet\to)$$

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

Note that $\gamma \cup \{X \preceq Y\}$ and $\gamma_1 \cup \gamma_2$ in the rules must be (valid) subtyping assumptions, i.e., any type variable must not have more than one occurrence in them. We also define a binary relation $\preceq$ over **TExp** as: $A \preceq B$ if and only if $\{\} \vdash A \preceq B$ is derivable.

Most of the subtyping rules are standard. The rule ($\preceq$-μ) corresponds to the "Amber rule" [7]. The rules ($\preceq$-$\top$), ($\preceq$-$\bullet$), ($\preceq$-approx), ($\preceq$-$\to\bullet$) and ($\preceq$-$\bullet\to$) reflect our intended meaning of the $\bullet$-modality discussed in Introduction.

Proposition 6.3. *1. If $\gamma \cup \{X \preceq Y\} \vdash A \preceq B$ is derivable, $X' \neq Y'$ and $\{X',\, Y'\} \cap FTV(\gamma) = \{\}$, then $\gamma \cup \{X' \preceq Y'\} \vdash A[X'/X, Y'/Y] \preceq B[X'/X, Y'/Y]$ is also derivable without changing the height of derivation.*

2. If $\gamma \cup \{X \preceq Y\} \vdash A \preceq B$ is derivable and $C \simeq D$, then $\gamma \vdash A[C/X, D/Y] \preceq B[C/X, D/Y]$ is also derivable without changing the height of derivation.

3. If $\gamma \vdash A \preceq B$ is derivable, and $\gamma \subset \gamma'$, then $\gamma' \vdash A \preceq B$ is also derivable without changing the height of derivation.

Proof. By simultaneous induction on the derivation of $\gamma \cup \{X \preceq Y\} \vdash A \preceq B$ or $\gamma \vdash A \preceq B$.

Proposition 6.4. *If* $\gamma \cup \{X \preceq Y\} \vdash A \preceq B$ *and* $\gamma \vdash C \preceq D$ *are derivable, then* $\gamma \vdash A[C/X, D/Y] \preceq B[C/X, D/Y]$ *is also derivable.*

Proposition 6.5. *Let* $\gamma \vdash A \preceq B$ *be a derivable subtyping judgment. If* $A \simeq \top$*, then* $B \simeq \top$.

Proof. By induction on the height of the derivation, and by cases of the last rule applied in the derivation. Some rules are impossible as the last one by Proposition 5.15. Use Proposition 5.16 for the case of ($\preceq$-$\bullet$). Use Proposition 5.9 for the case of ($\preceq$-approx). If the last rule is ($\preceq$-μ), then the derivation ends with:

$$\dfrac{\begin{array}{c}\vdots\\ \gamma \cup \{X \preceq Y\} \vdash A' \preceq B'\end{array}}{\gamma \vdash \mu X.A' \preceq \mu Y.B'}\ (\preceq\text{-}\mu)$$

for some X, Y, A' and B' such that $A = \mu X.A'$, $B = \mu Y.B'$, $X \notin FTV(B')$, $Y \notin FTV(A')$, and A' and B' are proper in X and Y, respectively. By Proposition 6.3.2, we can get a derivation of $\gamma \vdash A'[\top/X] \preceq B'[\top/Y]$ from the one of $\gamma \cup \{X \preceq Y\} \vdash A' \preceq B'$ without changing the height of derivation. Since $A \simeq A'[A/X]$ and $A \simeq \top$, we get $A'[\top/X] \simeq \top$. Therefore, by the induction hypothesis, $B'[\top/X] \simeq \top$; and then by ($\simeq$-uniq), $B = \mu Y.B' \simeq \top$. $\square$

Proposition 6.6. *Let* $\gamma \vdash A \preceq B$ *be a derivable subtyping judgment. If* $A \simeq \bullet A'$*, then* $B \simeq \bullet B'$ *for some* B' *such that* $\gamma \vdash A' \preceq B'$ *is derivable.*

Proof. Since $A' \simeq \top$ implies $B \simeq \top \simeq \bullet\top$ by Proposition 6.5, we assume that$A' \not\simeq \top$. The proof proceeds by induction on the derivation, and by the cases of the last rule applied in the derivation. Some rules are impossible as the last one by Proposition 5.15. Use Proposition 5.14.1 for the case of ($\preceq$-$\bullet$). If the last rule is ($\preceq$-μ), then the derivation ends with:

$$\dfrac{\begin{array}{c}\vdots\\ \gamma \cup \{X \preceq Y\} \vdash C \preceq D\end{array}}{\gamma \vdash \mu X.C \preceq \mu Y.D}\ (\preceq\text{-}\mu)$$

for some X, Y, C and D such that $A = \mu X.C$, $B = \mu Y.D$, $X \notin FTV(D)$, $Y \notin FTV(C)$, and C and D are proper in X and Y, respectively. Since $A' \not\simeq \top$, $A' \not\simeq X$ and $A \simeq C[A/X] \simeq \bullet A'$, by considering the canonical form of A', we get $C \simeq \bullet C'$ for some C' by Proposition 5.18. Therefore, by the induction hypothesis, $\gamma \cup \{X \preceq Y\} \vdash C' \preceq D'$ is derivable for some D' such that $D \simeq \bullet D'$. We get the derivation of $\gamma \vdash C'[A/X, B/Y] \preceq D'[A/X, /B/Y]$ by Proposition 6.4. On the other hand, since $Y \notin FTV(C)$ and $X \notin FTV(B)$, $\bullet A' \simeq A = \mu X. C = \mu X.C[B/Y] \simeq \mu X.\bullet C'[B/Y] \simeq \bullet C'[B/Y][A/X] = \bullet C'[A/X, B/Y]$. Therefore, $A' \simeq C'[A/X, B/Y]$ by Proposition 5.14.1. We thus get $\gamma \vdash A' \preceq D'[A/X, /B/Y]$. Note that $B = \mu Y.D = \mu Y.D[A/X] \simeq \mu Y.\bullet D'[A/X] \simeq \bullet D'[A/X][B/Y] \simeq \bullet D'[A/X, B/Y]$ since $X \notin FTV(D)$ and $Y \notin FTV(A)$. $\square$

Proposition 6.7. *Let* $\gamma \vdash A \preceq B$ *be a derivable subtyping judgment.*

1. *If* $A \simeq \bullet^m P$ *and* $B \not\simeq \top$*, then* $B \simeq \bullet^n P$ *for some* n *such that* $m \leq n$.
2. *If* $B \simeq \bullet^n P$*, then* $A \simeq \bullet^m P$ *for some* m *such that* $m \leq n$.

Proof. By induction on the derivation of $\gamma \vdash A \preceq B$, and by cases of the last rule applied in the derivation. In case of ($\preceq$-trans), use Proposition 6.5 to apply the induction hypothesis. Use Proposition 5.14.1 for the cases ($\preceq$-$\bullet$) and ($\preceq$-approx). If the last rule is ($\preceq$-μ), then the derivation ends with:

$$\dfrac{\begin{array}{c}\vdots\\ \gamma \cup \{X \preceq Y\} \vdash A' \preceq B'\end{array}}{\gamma \vdash \mu X.A' \preceq \mu Y.B'}\ (\preceq\text{-}\mu)$$

for some X, Y, A' and B' such that $A = \mu X.A'$, $B = \mu Y.B'$, $X \notin FTV(B')$, $Y \notin FTV(A')$, and A' and B' are proper in X and Y, respectively. If $A \simeq \bullet^m P$, then since $A \simeq A'[A/X] \simeq \bullet^m P$ and $A' \not\simeq X$, we get $A' \simeq \bullet^m P$ by Proposition 5.18; and therefore, by the induction hypothesis, $B' \simeq \bullet^n P$ for some $n \geq m$. Similarly, if $B \simeq \bullet^n P$, then $A' \simeq \bullet^m P$ for some $m \leq n$. $\square$

Proposition 6.8. *Let $\gamma \vdash A \preceq B$ be a derivable subtyping judgment.*

1. *If $A \simeq \bullet^m X$ and $B \not\simeq \top$, then $B \simeq \bullet^n Y$ for some n and Y such that: (a) $m \leq n$, and (b) $X = Y$ or $\{X \preceq Y\} \in \gamma$.*
2. *If $B \simeq \bullet^n Y$, then $A \simeq \bullet^m X$ for some m and X such that: (a) $m \leq n$, and (b) $X = Y$ or $\{X \preceq Y\} \in \gamma$.*

Proof. Similar to Proposition 6.7. □

Proposition 6.9. *Let $\gamma \vdash A \preceq B$ be a derivable subtyping judgment.*

1. *If $A \simeq \bullet^m(C \to D)$ and $B \not\simeq \top$, then there exist some n, k, l, E and F such that:*
   (a1) $B \simeq \bullet^n(E \to F)$,
   (a2) $m+k \leq n$, *and*
   (a3) $\gamma \vdash \bullet^k E \preceq \bullet^l C$ *and* $\gamma \vdash \bullet^l D \preceq \bullet^k F$ *are derivable.*
2. *If $B \simeq \bullet^n(E \to F)$, then there exist some m, k, l, C and D such that:*
   (b1) $A \simeq \bullet^m(C \to D)$,
   (b2) $m+k \leq n$,
   (b3) $\gamma \vdash \bullet^l D \preceq \bullet^k F$ *is derivable, and*
   (b4) so is $\gamma \vdash \bullet^k E \preceq \bullet^l C$ *provided* $F \not\simeq \top$.

Note that k, l, m and n range over non-negative integers.

Proof. By induction on the derivation of $\gamma \vdash A \preceq B$, and by cases of the last rule used in the derivation. Proposition 5.18 is crucial to the case ($\preceq$-μ). For the first part of the conjecture, suppose that $A \simeq \bullet^m(C \to D)$ and $B \not\simeq \top$.

Case: ($\preceq$-*assump*). This case is impossible by Proposition 5.15.

Case: ($\preceq$-$\top$). Also impossible Proposition 6.5, since $B \not\simeq \top$.

Case: ($\preceq$-*reflex*). Trivial because $B \simeq A \simeq \bullet^m(C \to D)$ in this case.

Case: ($\preceq$-*trans*). In this case, $\gamma \vdash A \preceq G$ and $\gamma \vdash G \preceq B$ are derivable for some G. We get $G \not\simeq \top$ from $B \not\simeq \top$ by Proposition 6.5. By the induction hypothesis on $\gamma \vdash A \preceq G$, there exist some n', k', l', E' and F' such that

- $G \simeq \bullet^n(E' \to F')$,
- $m+k' \leq n'$, and
- $\gamma \vdash \bullet^{k'} E \preceq \bullet^{l'} C$ and $\gamma \vdash \bullet^{l'} D \preceq \bullet^{k'} F$ are derivable.

Therefore, by the induction hypothesis on $\gamma \vdash G \preceq B$, there exist some n, k'', l'', E and F such that

- $B \simeq \bullet^n(E \to F)$,
- $n'+k'' \leq n$, and
- $\gamma \vdash \bullet^{k''} E \preceq \bullet^{l''} E'$ and $\gamma \vdash \bullet^{l''} F' \preceq \bullet^{k''} F$ are derivable.

Taking k and l as $k = k' + k''$ and $l = l' + l''$, we get (a1) through (a3).

Case: ($\preceq$-$\bullet$). In this case, $\gamma \vdash A' \preceq B'$ are derivable for some A' and B' such that $A = \bullet A'$ and $B = \bullet B'$. We get $B' \not\simeq \top$ from $B \not\simeq \top$ by Proposition 5.9, and $m > 0$ and $A' \simeq \bullet^{m-1}(C \to D)$ from $A \simeq \bullet^m(C \to D)$ by Propositions 5.14.1 and 5.15. Therefore, by the induction hypothesis, there exist some n', k, l, E and F such that:

- $B' \simeq \bullet^{n'}(E \to F)$,
- $m-1+k \leq n'$, and
- $\gamma \vdash \bullet^k E \preceq \bullet^l C$ and $\gamma \vdash \bullet^l D \preceq \bullet^k F$ are derivable.

It then suffices to take n as $n = n' + 1$.

Case: ($\preceq$-$\rightarrow$). In this case, $\gamma \vdash B_1 \preceq A_1$ and $\gamma \vdash A_2 \preceq B_2$ are derivable for some A_1, A_2, B_1 and B_2 such that $A = A_1 \rightarrow A_2$ and $B = B_1 \rightarrow B_2$. Since $A \simeq \bullet^m(C \rightarrow D)$, we get $m = 0$ by Proposition 5.15, and

- $D \simeq A_2 \simeq \top$, or
- $C \simeq A_1$ and $D \simeq A_2$

by Propositions 5.14.2. In the former case, we get $B_2 \simeq \top$ from $\gamma \vdash A_2 \preceq B_2$ by Proposition 6.5; and hence, $B = B_1 \rightarrow B_2 \simeq C \rightarrow D$ by ($\simeq$-$\rightarrow\top$). Therefore, taking n, k, E, F and l as $n = 0$, $k = 0$, $E = C$, $F = D$ and $l = 0$, we get (a1) through (a3) in this case. In the latter case, it suffices to take n, k, l, E and F as $n = 0$, $k = 0$, $l = 0$, $E = B_1$ and $F = B_2$.

Case: ($\preceq$-*approx*). Since $B = \bullet A$ in this case, it suffices to take n, k, l, E and F as $n = m + 1$, $k = 0$, $l = 0$, $E = C$ and $F = D$.

Case: ($\preceq$-$\rightarrow\bullet$). In this case, $A = A_1 \rightarrow A_2$ and $B = \bullet A_1 \rightarrow \bullet A_2$ for some A_1 and A_2. Since $A \simeq \bullet^m(C \rightarrow D)$, we get $m = 0$ by Proposition 5.15, and

- $D \simeq A_2 \simeq \top$, or
- $C \simeq A_1$ and $D \simeq A_2$

by Propositions 5.14.2. In the former case, $B = \bullet A_1 \rightarrow \bullet A_2 \simeq \bullet A_1 \rightarrow \bullet\top \simeq \bullet A_1 \rightarrow \top \simeq C \rightarrow D$ by ($\simeq$-$\rightarrow\top$) and Proposition 5.9. Therefore, taking n, k, l, E and F as $n = 0$, $k = 0$, $l = 0$, $E = C$ and $F = D$, we get (a1) through (a3) in this case. In the latter case, it suffices to take n, k, l, E and F as $n = 0$, $k = 0$, $l = 1$, $E = \bullet C$ and $F = \bullet D$.

Case: ($\preceq$-$\bullet\rightarrow$). In this case, $A = \bullet A_1 \rightarrow \bullet A_2$ and $B = \bullet(A_1 \rightarrow A_2)$ for some A_1 and A_2. Since $A \simeq \bullet^m(C \rightarrow D)$, we get $m = 0$ by Proposition 5.15, and

- $D \simeq \bullet A_2 \simeq \top$, or
- $C \simeq \bullet A_1$ and $D \simeq \bullet A_2$

by Propositions 5.14.2. In the former case, $A_2 \simeq \top$ by Proposition 5.16, and hence, $B = \bullet(A_1 \rightarrow A_2) \simeq \bullet(A_1 \rightarrow \top) \simeq \bullet(C \rightarrow D)$ by ($\simeq$-$\rightarrow\top$). Therefore, taking n, k, l, E and F as $n = 1$, $k = 0$, $l = 0$, $E = C$ and $F = D$, we get (a1) through (a3) in this case. In the latter case, it suffices to take n, k, l, E and F as $n = 1$, $k = 1$, $l = 0$, $E = A_1$ and $F = A_2$.

Case: ($\preceq$-μ). In this case, $\gamma \cup \{X \preceq Y\} \vdash A' \preceq B'$ is derivable for some X, Y, A' and B' such that:

$$A = \mu X.A', \tag{8}$$
$$B = \mu Y.B', \tag{9}$$
$$X \notin FTV(\gamma) \cup FTV(B'), \quad \text{and} \tag{10}$$
$$Y \notin FTV(\gamma) \cup FTV(A'). \tag{11}$$

Since A' is proper in X, we get $A' \not\simeq X$. Note that $\bullet^m(C \rightarrow D) \simeq A = \mu X.A' \simeq A'[A/X]$ by (8) and ($\simeq$-fix). Therefore, by Proposition 5.18, there exist some C' and D' such that:

$$A' \simeq \bullet^m(C' \rightarrow D') \tag{12}$$
$$D \simeq D'[A/X], \text{ and} \tag{13}$$
$$C \simeq C'[A/X] \text{ or } D \simeq D'[A/X] \simeq \top, \tag{14}$$

Since we can get $B' \not\simeq \top$ from $B \not\simeq \top$. By the induction hypothesis, there exist some n, k', l', E' and F' such that:

(a1′) $B' \simeq \bullet^n(E' \rightarrow F')$,
(a2′) $m+k' \leq n$, and
(a3′) $\gamma \cup \{X \preceq Y\} \vdash \bullet^{k'}E' \preceq \bullet^{l'}C'$ and $\gamma \cup \{X \preceq Y\} \vdash \bullet^{l'}D' \preceq \bullet^{k'}F'$ are derivable.

There are two subcases as follows.

Subcase: $D'[A/X] \simeq \top$. Since $A \simeq \bullet^m(C \to D)$ implies $A \not\simeq \top$ by Proposition 5.15, we get

$$D' \simeq \top, \tag{15}$$

by Proposition 4.9.2; and therefore,

$$F' \simeq \top \tag{16}$$

from (a3′) by Propositions 6.5 and 5.16. Hence,

$$\begin{array}{lll}
B & \simeq B'[B/Y] & \text{(by (9) and } (\simeq\text{-fix}))\\
& \simeq (\bullet^n(E' \to F'))[B/Y] & \text{(by (a1}'))\\
& \simeq (\bullet^n(E' \to \top))[B/Y] & \text{(by (16))}\\
& \simeq (\bullet^n(C' \to \top))[B/Y] & \text{(by } (\simeq\text{-}{\to}\top))\\
& \simeq (\bullet^n(C' \to D'))[B/Y] & \text{(by (15))}\\
& \simeq (\bullet^{n-m}A')[B/Y] & \text{(by (12))}\\
& = \bullet^{n-m}A' & \text{(by (11))}\\
& \simeq \bullet^n(C \to D) & \text{(by (12))}
\end{array}$$

Therefore, in this subcase, we get (a1) through (a3) by taking k, l, E and F as $k = 0$ $l = 0$, $E = C$ and $F = D$.

Subcase: $D'[A/X] \not\simeq \top$. In this case, obviously $D' \not\simeq \top$, and we get

$$C \simeq C'[A/X] \tag{17}$$

from (14). Let E and F be as $E = E'[A/X,\, B/Y]$ and $F = F'[A/X,\, B/Y]$.

$$\begin{array}{lll}
B & \simeq B'[B/Y] & \text{(by (9) and } (\simeq\text{-fix}))\\
& = B'[A/X,\, B/Y] & \text{(by (10))}\\
& \simeq (\bullet^n(E' \to F'))[A/X,\, B/Y] & \text{(by (a1}'))\\
& = \bullet^n(E \to F) &
\end{array}$$

Since $D' \not\simeq \top$, and since $\bullet^m(C'[B/Y] \to D'[B/Y]) = \bullet^m(C' \to D')[B/Y] \simeq A'[B/Y] = A' \simeq \bullet^m(C' \to D')$ by (11) and (12), we get $C'[B/Y] \simeq C'$ and $D'[B/Y] \simeq D'$ by Proposition 5.14; and therefore,

$$\begin{array}{l}
C'[A/X,\, B/Y] = C'[B/Y][A/X] \simeq C'[A/X] \simeq C, \text{ and}\\
D'[A/X,\, B/Y] = D'[B/Y][A/X] \simeq D'[A/X] \simeq D
\end{array}$$

by (10), (13) and (17). Hence, taking k and l as $k = k'$ and $l = l'$, $\gamma \vdash \bullet^k E \preceq \bullet^l C$ and $\gamma \vdash \bullet^l D \preceq \bullet^k F$ are derivable, because so are $\gamma \vdash \bullet^k E'[A/X,\, B/Y] \preceq \bullet^l C'[A/X,\, B/Y]$ and $\gamma \vdash \bullet^l D'[A/X,\, B/Y] \preceq \bullet^k F'[A/X,\, B/Y]$ by (a3′) and Proposition 6.4. We thus get (a1) through (a3) also in this subcase.

We can prove the second part of the conjecture similarly. □

Proposition 6.10. *Let n and m be non-negative integers.*

1. $\bullet^n\top \not\preceq \bullet^m P$, $\bullet^n\top \not\preceq \bullet^m X$ *and* $\bullet^n\top \not\preceq \bullet^m(A \to B)$.
2. $\bullet^n P \not\preceq \bullet^m X$ *and* $\bullet^n P \not\preceq \bullet^m(A \to B)$.
3. $\bullet^n X \not\preceq \bullet^m P$ *and* $\bullet^n X \not\preceq \bullet^m(A \to B)$.
4. $\bullet^n(A \to B) \not\preceq \bullet^m P$ *and* $\bullet^n(A \to B) \not\preceq \bullet^m X$.
5. $\bullet A \not\preceq P$, $\bullet A \not\preceq X$, *and* $\bullet A \not\preceq B \to C$.

Proof. Obvious from Propositions 6.6, 6.7, 6.8, 6.9, 5.9 and 5.15. □

Proposition 6.11. *If* $\gamma \vdash \bullet^n A \preceq A$ *is derivable and* $A \not\simeq \top$, *then* $n = 0$.

Proof. Straightforward from Propositions 6.7, 6.8 and 6.9 since $\gamma \vdash \bullet^n A^c \preceq A^c$ is also derivable and $A^c \not\simeq \top$.

7 The typing rules

We now define the typing rules of $\lambda\bullet\mu$. According to the intended meaning of $\bullet$, two new typing rules, $(\bullet)$ and $(\preceq)$, are added and the $(\to\mathrm{E}_{\lambda\mu})$ rule is generalized to handle types with the $\bullet$-modality.

Definition 7.1 (Typing rules). Typing contexts for $\lambda\bullet\mu$ are defined similarly to the case of $\lambda\mu$. Let τ be a mapping that assigns a type constant $\tau(c)$ to each individual constant c. The typing system $\lambda\bullet\mu$ is defined relatively to τ by the following derivation rules:

$$\frac{}{\Gamma \cup \{x : A\} \vdash x : A}\ (\text{var}) \qquad\qquad \frac{}{\Gamma \vdash c : \tau(c)}\ (\text{const})$$

$$\frac{}{\Gamma \vdash M : \top}\ (\top) \qquad \frac{\bullet\Gamma \vdash M : \bullet A}{\Gamma \vdash M : A}\ (\bullet) \qquad \frac{\Gamma \vdash M : A \qquad \vdash A \preceq B}{\Gamma \vdash M : B}\ (\preceq)$$

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

where $\bullet\,\Gamma$ denotes the typing context $\{x_1 : \bullet A_1,\, x_2 : \bullet A_2,\, \ldots,\, x_n : \bullet A_n\}$ when $\Gamma = \{\, x_1 : A_1,\, x_2 : A_2,\, \ldots,\, x_n : A_n\}$.

The $(\bullet)$-rule represents the fact that every possible world n has its successor $n{+}1$. Since the interpretation of $\Gamma \vdash M : A$ in the world n is identical to the one of $\bullet\Gamma \vdash M : \bullet A$ in the world $n{+}1$, $\Gamma \vdash M : A$ is valid whenever so is $\bullet\Gamma \vdash M : \bullet A$. The $(\to\mathrm{E})$-rule allows us to derive $\vdash \lambda x.\, \lambda y.\, xy : \bullet(A \to B) \to \bullet A \to \bullet B$ for every A and B. Therefore, $\bullet(A \to B)$ and $\bullet A \to \bullet B$ are *logically* equivalent, even though not equivalent as sets of λ-terms. Note that $\bullet A \to \bullet B \preceq \bullet(A \to B)$, but $\bullet(A \to B) \not\preceq \bullet A \to \bullet B$.

Example 7.2. We can derive Curry's fixed-point combinator **Y** in $\lambda\bullet\mu$; more precisely, the following is derivable.

$$\vdash \lambda f.\, (\lambda x.\, f(xx))\, (\lambda x.\, f(xx)) \;:\; (\bullet X \to X) \to X$$

Let a formula $A = \mu Y.\bullet Y \to X$ and a derivation Π as follows:

$$\Pi = \cfrac{\cfrac{\cfrac{}{f : \bullet X \to X \vdash f : \bullet X \to X}\ (\text{var}) \qquad \cfrac{\cfrac{\cfrac{}{x : \bullet A \vdash x : \bullet A}\ (\text{var})}{x : \bullet A \vdash x : \bullet(\bullet A \to X)}\ (\simeq) \qquad \cfrac{\cfrac{}{x : \bullet A \vdash x : \bullet A}\ (\text{var})}{x : \bullet A \vdash x : \bullet\bullet A}\ (\preceq)}{x : \bullet A \vdash xx : \bullet X}\ (\to\mathrm{E})}{f : \bullet X \to X,\, x : \bullet A \vdash f(xx) : X}\ (\to\mathrm{E})}{f : \bullet X \to X \vdash \lambda x.\, f(xx) : \bullet A \to X}\ (\to\mathrm{I})$$

Then, we can derive **Y** as follows:

$$\cfrac{\cfrac{\begin{array}{c}\vdots\ \Pi\\ f : \bullet X \to X \vdash \lambda x.\, f(xx) : \bullet A \to X\end{array} \qquad \cfrac{\begin{array}{c}\vdots\ \Pi\\ f : \bullet X \to X \vdash \lambda x.\, f(xx) : \bullet A \to X\end{array}}{f : \bullet X \to X \vdash \lambda x.\, f(xx) : \bullet A}\ (\preceq)}{f : \bullet X \to X \vdash (\lambda x.\, f(xx))\, (\lambda x.\, f(xx)) : X}\ (\to\mathrm{E})}{\vdash \lambda f.\, (\lambda x.\, f(xx))\, (\lambda x.\, f(xx)) : (\bullet X \to X) \to X}\ (\to\mathrm{I})$$

We can also observe that Turing's fixed point combinator $(\lambda x.\lambda f.f(xxf))\,(\lambda x.\,\lambda f.\, f(xxf))$ has the same type. The type $(\bullet X \to X) \to X$ gives a concise axiomatic meaning to the fixed point combinators; it says that they can produce an element of X with a given function that works as an information pump from $\bullet X$ to X; in other words, they provide the induction scheme discussed in Introduction. The type thus enables us to construct recursive programs using the fixed point combinators without analyzing their computational behavior. We will see some examples of such recursive programs in Section 14.

8 Basic properties of $\boldsymbol{\lambda\bullet\mu}$

The typing system $\lambda\bullet\mu$ enjoys some basic properties such as subject reduction property.

Proposition 8.1. *If* $\Gamma \vdash M : B$ *is derivable, then* $FV(M) \subset Dom(\Gamma)$.

Proof. By straightforward induction on the derivation of . $\Gamma \vdash M : B$. □

Proposition 8.2. *Let* $\Gamma = \{x_1 : A_1, x_2 : A_2, \ldots, x_n : A_n\}$ *and* $\Gamma' = \{x_1 : A'_1, x_2 : A'_2, \ldots, x_n : A'_n\}$. *If* $\Gamma \vdash M : B$ *is derivable,* $B \preceq B'$, *and* $A'_i \preceq A_i$ *for every* i $(1 \le i \le n)$, *then* $\Gamma' \vdash M : B'$ *is also derivable.*

Lemma 8.3. *Let* Γ_1 *and* Γ_2 *be typing contexts such that* $Dom(\Gamma_1) \cap Dom(\Gamma_2) = \{\}$. *If* $\Gamma_1 \cup \Gamma_2 \vdash M : A$ *is derivable, then so is* $\bullet\Gamma_1 \cup \Gamma_2 \vdash M : \bullet A$.

Proof. By induction on the derivation. Use the property $(\preceq$-$\bullet\to)$-rule for the case that the last rule is $(\to\mathrm{I})$. □

Lemma 8.4 (Substitution lemma). *If* $\Gamma \cup \{x : A\} \vdash M : B$ *and* $\Gamma \vdash N : A$ *are derivable, then so is* $\Gamma \vdash M[N/x] : B$.

Proof. By induction on the derivation of $\Gamma_1 \cup \{x : A\} \vdash M : B$, and by cases of the last rule. Use Lemma 8.3 when the last rule is $(\bullet)$. □

Theorem 8.5 (Subject reduction). *If* $\Gamma \vdash M : A$ *is derivable and* $M \underset{\beta}{\to} M'$, *then* $\Gamma \vdash M' : A$ *is derivable.*

Proof. By induction on the structure of M. Suppose that $M \underset{\beta}{\to} M'$. We show that $\Gamma \vdash M' : A$ by cases of the form of M. Most of the cases are straight forward. In the case that $M = M_1 M_2$ for some M_1 and M_2, the derivation of $\Gamma \vdash M : A$ must end with:

$$\dfrac{\dfrac{}{\Gamma \vdash M_1 M_2 : \top}\,(\top) \quad \vdots \text{ zero or more } (\preceq)}{\Gamma \vdash M_1 M_2 : A}\,(\preceq) \qquad \text{or} \qquad \dfrac{\dfrac{\dfrac{\vdots}{\Gamma \vdash M_1 : \bullet^n(B \to C)} \quad \dfrac{\vdots}{\Gamma \vdash M_2 : \bullet^n B}}{\Gamma \vdash M_1 M_2 : \bullet^n C}\,(\to\mathrm{E}) \quad \vdots \text{ zero or more } (\preceq)}{\Gamma \vdash M_1 M_2 : A}\,(\preceq)$$

for some n, B and C such that $\bullet^n C \preceq A$. In the former case, $\Gamma \vdash M' : A$ is derivable by $(\top)$ and $(\preceq)$. So we concentrate on the latter case. If $C \simeq \top$, then $\Gamma \vdash M' : A$ is derivable by $(\top)$ and $(\preceq)$. Therefore, we also assume that $C \simeq \top$ in the sequel. Since $M_1 M_2 \underset{\beta}{\to} M'$, there are three possible cases as follows:

(a) $M_1 \overset{*}{\underset{\beta}{\to}} M'_1$ and $M' = M'_1 M_2$ for some M'_1.
(b) $M_2 \overset{*}{\underset{\beta}{\to}} M'_2$ and $M' = M_1 M'_2$ for some M'_2.
(c) $M_1 = \lambda y.\, L$ and $M' = L[M_2/y]$ for some y and L.

By the induction hypothesis, we have the derivations of $\Gamma \vdash M'_1 : \bullet^n(B \to C)$ and $\Gamma \vdash M'_2 : \bullet^n B$ for (a) and (b), respectively. Therefore, we get the derivation of $\Gamma \vdash M' : \bullet^n C$ by applying $(\to\mathrm{E})$, and then $\Gamma \vdash M' : A$ by $(\preceq)$ in these cases. As for (c), since $\top \not\preceq \bullet^n(B \to C)$, the derivation of $\Gamma \vdash \lambda y.\, L : \bullet^n(B \to C)$ must end with:

$$\dfrac{\dfrac{\dfrac{\vdots}{\Gamma \cup \{y : D\} \vdash L : E}}{\Gamma \vdash \lambda y.\, L : D \to E}\,(\to\mathrm{I}) \quad \vdots \text{ zero or more } (\preceq)}{\Gamma \vdash \lambda y.\, L : \bullet^n(B \to C)}$$

for some D and E such that $D \to E \preceq \bullet^n(B \to C)$. Since we now have the assumption $C \not\simeq \top$, by Proposition 6.9, there exist some m, k, l, D' and E' such that:

- $D \to E \simeq \bullet^m(D' \to E')$,

– $m+k \leq n$,
– $\bullet^l E' \preceq \bullet^k C$ and $\bullet^k B \preceq \bullet^l D'$.

Obviously $m = 0$ by Proposition 5.15; and $D \simeq D'$ and $E \simeq E'$ by Proposition 5.14.2 because we get $E' \not\simeq \top$ from $C \not\simeq \top$ and $\bullet^l E' \preceq \bullet^k C$. Therefore,

$$\bullet^n B \preceq \bullet^{l+n-k} D' \simeq \bullet^{l+n-k} D, \text{ and} \tag{18}$$

$$\bullet^{l+n-k} E \simeq \bullet^{l+n-k} E' \preceq \bullet^n C. \tag{19}$$

On the other hand, we get a derivation of $\Gamma \cup \{y : \bullet^{l+n-k} D\} \vdash L : \bullet^{l+n-k} E$ from the one of $\Gamma \cup \{y : D\} \vdash L : E$ by Lemma 8.3. Since $\Gamma \vdash M_2 : \bullet^n B$ is derivable, so is $\Gamma \vdash M_2 : \bullet^{l+n-k} D$ by (18). We so get $\Gamma \vdash L[M_2/x] : \bullet^{l+n-k} E$ by Lemma 8.4, and then, $\Gamma \vdash L[M_2/x] : \bullet^n C$ by (19). Finally, we get $\Gamma \vdash L[M_2/x] : A$ by applying $(\preceq)$. □

9 Interpretations of types

Definition 9.1. Let $<\mathcal{T}, \sqsubseteq>$ be a partially ordered set. We define a set $\mathcal{A}(\mathcal{T}, \sqsubseteq)$ of infinite sequences of elements of $\mathcal{T}$ as follows:

$$\mathcal{A}(\mathcal{T}, \sqsubseteq) = \{ <t_0, t_1, t_2, \ldots, t_k, \ldots> \mid t_{k+1} \sqsubseteq t_k \in \mathcal{T} \text{ for every } k \}.$$

We denote the k-th element of $t \in \mathcal{A}(\mathcal{T}, \sqsubseteq)$ by t_k. Note that t starts with its 0-th element t_0. We extend $\sqsubseteq$ to $\mathcal{A}(\mathcal{T}, \sqsubseteq)$ as:

$$t \sqsubseteq s \text{ if and only if } t_k \sqsubseteq s_k \text{ for every } k.$$

We also define $\sqsubseteq_k$, $\sqsubseteq_{<k}$, $\sqsubseteq_{\leq k}$, $=_k$, $=_{<k}$ and $=_{\leq k}$ as follows:

$$\begin{aligned}
t \sqsubseteq_k s &\text{ if and only if } t_k \sqsubseteq s_k \\
t \sqsubseteq_{<k} s &\text{ if and only if } t_l \sqsubseteq s_l \text{ for every } l < k \\
t \sqsubseteq_{\leq k} s &\text{ if and only if } t_l \sqsubseteq s_l \text{ for every } l \leq k \\
t =_k s &\text{ if and only if } t_k = s_k \\
t =_{<k} s &\text{ if and only if } t_l = s_l \text{ for every } l < k \\
t =_{\leq k} s &\text{ if and only if } t_l = s_l \text{ for every } l \leq k
\end{aligned}$$

Definition 9.2. Let $<\mathcal{T}, \sqsubseteq>$ is a partially ordered set with the least element $\bot_{\mathcal{T}}$. Let $t = <t_0, t_1, t_2, \ldots>$ be an element of $\mathcal{A}(\mathcal{T}, \sqsubseteq)$, and n a non-negative integer. We define an element $t|_{<n}$ of $\mathcal{A}(\mathcal{T}, \sqsubseteq)$ as follows:

$$(t|_{<n})_k = \begin{cases} t_k & (\text{if } k < n) \\ \bot_{\mathcal{T}} & (\text{if } n \leq k) \end{cases}$$

Definition 9.3. Let $<\mathcal{T}, \sqsubseteq>$ is a partially ordered set with the greatest element $\top_{\mathcal{T}}$. We define an element $\top_{\mathcal{A}(\mathcal{T}, \sqsubseteq)}$ of $\mathcal{A}(\mathcal{T}, \sqsubseteq)$ as follows:

$$(\top_{\mathcal{A}(\mathcal{T}, \sqsubseteq)})_k = \top_{\mathcal{T}}.$$

We often write $\top_{\mathcal{A}}$ instead of $\top_{\mathcal{A}(\mathcal{T}, \sqsubseteq)}$ when $\mathcal{T}$ and $\sqsubseteq$ is clear from the context.

Definition 9.4 (Interpretations of types). An *interpretation of types* is a tuple $<\mathcal{T}, \sqsubseteq, \theta, \boxed{\bullet}, \boxed{\rightarrow}>$ such that:

(i1) $<\mathcal{T}, \sqsubseteq>$ is a partially ordered set with the least element $\bot_{\mathcal{T}}$ and the greatest element $\top_{\mathcal{T}}$.
(i2) $\theta : \textbf{TConst} \to \mathcal{A}(\mathcal{T}, \sqsubseteq)$
(i3) $\boxed{\bullet} : \mathcal{A}(\mathcal{T}, \sqsubseteq) \to \mathcal{A}(\mathcal{T}, \sqsubseteq)$
(i4) If $t =_{<k} s$ then $\boxed{\bullet}(t) =_{\leq k} \boxed{\bullet}(s)$.
(i5) $- \boxed{\rightarrow} - : \mathcal{A}(\mathcal{T}, \sqsubseteq) \times \mathcal{A}(\mathcal{T}, \sqsubseteq) \to \mathcal{A}(\mathcal{T}, \sqsubseteq)$
(i6) If $t' =_{\leq k} t$ and $s =_{\leq k} s'$, then $t \boxed{\rightarrow} s =_{\leq k} t' \boxed{\rightarrow} s'$.
(t1) $\boxed{\bullet}(\top_{\mathcal{A}}) = \top_{\mathcal{A}}$.

(t2) $t \boxed{\rightarrow} \top_{\mathcal{A}} = s \boxed{\rightarrow} \top_{\mathcal{A}}$.
(s1) If $t \sqsubseteq_{<k} s$ then $\boxed{\bullet}(t) \sqsubseteq_{\le k} \boxed{\bullet}(s)$.
(s2) If $t' \sqsubseteq_{\le k} t$ and $s \sqsubseteq_{\le k} s'$, then $t \boxed{\rightarrow} s \sqsubseteq_{\le k} t' \boxed{\rightarrow} s'$.
(s3) $t \sqsubseteq \boxed{\bullet}(t)$.
(s4) $t \boxed{\rightarrow} s \sqsubseteq \boxed{\bullet}(t) \boxed{\rightarrow} \boxed{\bullet}(s)$.
(s5) $\boxed{\bullet}(t) \boxed{\rightarrow} \boxed{\bullet}(s) \sqsubseteq \boxed{\bullet}(t \boxed{\rightarrow} s)$.

The conditions (i4) and (i6) are redundant, since (s1) and (s2) imply (i4) and (i6), respectively. Note also that the condition (i4) implies $\boxed{\bullet}(t)_0 = \boxed{\bullet}(s)_0$ for every $t, s \in \mathcal{A}(\mathcal{T}, \sqsubseteq)$.

Definition 9.5. Let $\mathcal{I} = <\mathcal{T}, \sqsubseteq, \theta, \boxed{\bullet}, \boxed{\rightarrow}>$ be an interpretation of types. We call a mapping $\xi : \mathbf{TVar} \to \mathcal{A}(\mathcal{T}, \sqsubseteq)$ a *type environment*.

Definition 9.6 (Semantics of types). Let $\mathcal{I} = <\mathcal{T}, \sqsubseteq, \theta, \boxed{\bullet}, \boxed{\rightarrow}>$ be an interpretation of types. We define a mapping $\overline{\mathcal{I}} : \mathbf{TExp} \times (\mathbf{TVar} \to \mathcal{A}(\mathcal{T}, \sqsubseteq)) \to \mathcal{A}(\mathcal{T}, \sqsubseteq)$ as follows:

$$\begin{aligned}
\overline{\mathcal{I}}(P, \xi) &= \theta(P) \\
\overline{\mathcal{I}}(X, \xi) &= \xi(X) \\
\overline{\mathcal{I}}(\bullet A, \xi) &= \boxed{\bullet}(\overline{\mathcal{I}}(A, \xi)) \\
\overline{\mathcal{I}}(A \to B, \xi) &= \begin{cases} \top_{\mathcal{A}} \boxed{\rightarrow} \top_{\mathcal{A}} & \text{(if B is a $\top$-variant)} \\ \overline{\mathcal{I}}(A, \xi) \boxed{\rightarrow} \overline{\mathcal{I}}(B, \xi) & \text{(otherwise)} \end{cases} \\
\overline{\mathcal{I}}(\mu X.A, \xi) &= \overline{\mathcal{I}}(A[\mu X.A/X], \xi)
\end{aligned}$$

Note that the $\overline{\mathcal{I}}(A, \xi)_k$ is defined by induction on the lexicographic ordering of $<k, r(A)>$ because $\overline{\mathcal{I}}(\bullet A, \xi)_k$ only depends on $\overline{\mathcal{I}}(A, \xi)_l$ $(l < k)$ by Definition 9.4 (i4).

Proposition 9.7. *Let $\mathcal{I} = <\mathcal{T}, \sqsubseteq, \theta, \boxed{\bullet}, \boxed{\rightarrow}>$ be an interpretation of types, and ξ a type environment.*

1. *If $X \notin FTV(A)$, then $\overline{\mathcal{I}}(A, \xi) = \overline{\mathcal{I}}(A, \xi[t/X])$ for every $t \in \mathcal{A}(\mathcal{T}, \sqsubseteq)$.*
2. *$\overline{\mathcal{I}}(A[B/X], \xi) = \overline{\mathcal{I}}(A, \xi[\overline{\mathcal{I}}(B, \xi)/X])$.*

Proof. It suffices to prove the following:

1′. If $X \notin FTV(A)$, then $\overline{\mathcal{I}}(A, \xi) =_{\le k} \overline{\mathcal{I}}(A, \xi[t/X])$ for every k and $t \in \mathcal{A}(\mathcal{T}, \sqsubseteq)$.
2′. $\overline{\mathcal{I}}(A[B/X], \xi) =_{\le k} \overline{\mathcal{I}}(A, \xi[\overline{\mathcal{I}}(B, \xi)/X])$ for every k.

The proof proceeds by straightforward induction the lexicographic ordering of $<k, r(A)>$ using Definitions 9.4 and 9.6. In this proof we need only the conditions (i1) through (i6) of Definition 9.4. ☐

Proposition 9.8. *Let $\mathcal{I} = <\mathcal{T}, \sqsubseteq, \theta, \boxed{\bullet}, \boxed{\rightarrow}>$ be an interpretation of types, and ξ a type environment. If*

(a) $t =_{<k} s$ and A is proper in X, or
(b) $t =_{\le k} s$,

then $\overline{\mathcal{I}}(A, \xi[t/X]) =_{\le k} \overline{\mathcal{I}}(A, \xi[s/X])$.

Proof. By induction on the lexicographic ordering of $<k, r(A)>$, and by cases of the form of A. We need only the conditions (i1) through (i6) of Definition 9.4. Suppose that (a) or (b) holds.

Case: $A = P$. Obvious since $\overline{\mathcal{I}}(P, \xi)$ does not depend on ξ by Definition 9.6

Case: $A = Y$. If $Y \neq X$, it is similar to the previous case. Otherwise, i.e., if $Y = X$, then A is not proper in X. Therefore, we get $\overline{\mathcal{I}}(A, \xi[s/X]) = s =_{\le k} t = \overline{\mathcal{I}}(A, \xi[t/X])$ from (b).

Case: $A = \bullet A'$. Note that A is proper in X in this case. Since $t =_{<k} s$ by (a) or (b), we get $\overline{\mathcal{I}}(A', \xi[t/X]) =_{<k} \overline{\mathcal{I}}(A', \xi[s/X])$ by the induction hypothesis. Hence, $\overline{\mathcal{I}}(\bullet A', \xi[t/X]) = \boxed{\bullet}(\overline{\mathcal{I}}(A', \xi[t/x])) =_{\le k} \boxed{\bullet}(\overline{\mathcal{I}}(A', \xi[s/X])) = \overline{\mathcal{I}}(\bullet A', \xi[s/X])$ by Definitions 9.4 (i4) and 9.6.

Case: $A = A_1 \to A_2$. If A_2 is a $\top$-variant, then trivial from Definition 9.6. Otherwise, since $r(A_1), r(A_2) < r(A)$, we get $\overline{\mathcal{I}}(A_1, \xi[t/X]) =_{\leq k} \overline{\mathcal{I}}(A_1, \xi[s/X])$ and $\overline{\mathcal{I}}(A_2, \xi[t/X]) =_{\leq k} \overline{\mathcal{I}}(A_2, \xi[s/X])$ by the induction hypothesis. Therefore, $\overline{\mathcal{I}}(A_1 \to A_2, \xi[t/X]) = \overline{\mathcal{I}}(A_1, \xi[t/X]) \boxed{\to} \overline{\mathcal{I}}(A_2, \xi[t/X]) =_{\leq k} \overline{\mathcal{I}}(A_1, \xi[s/X]) \boxed{\to} \overline{\mathcal{I}}(A_2, \xi[s/X]) = \overline{\mathcal{I}}(A_1 \to A_2, \xi[s/X])$ by Definitions 9.4 (i6) and 9.6.

Case: $A = \mu Y.A'$. We can assume that $Y \notin \{X\}$ without loss of generality. Note that $r(A'[\mu Y.A'/Y]) < r(\mu Y.A')$ by Proposition 4.6; and $A'[\mu Y.A'/Y]$ is proper in X if and only if so is $\mu Y.A'$, by Proposition 5.5.2.

$$\begin{array}{rcll} \overline{\mathcal{I}}(\mu Y.A', \xi[t/X]) & = & \overline{\mathcal{I}}(A'[\mu Y.A'/Y], \xi[t/X]) & \text{(by Definition 9.6)} \\ & =_{\leq k} & \overline{\mathcal{I}}(A'[\mu Y.A'/Y], \xi[s/X]) & \text{(by the induction hypothesis)} \\ & = & \overline{\mathcal{I}}(\mu Y.A', \xi[s/X]) & \text{(by Definition 9.6)} \end{array}$$

□

Remark 9.9. By considering the following distance function over $\mathcal{A}(\mathcal{T}, \sqsubseteq)$, we obtain a complete metric space of types.

$$d(<s_0, s_1, s_2, \ldots>, <t_0, t_1, t_2, \ldots>) = \begin{cases} 0 & (s_k = t_k \text{ for every } k) \\ 2^{-\min\{\, k \mid s_k \neq t_k \,\}} & \text{(otherwise)} \end{cases}$$

Proposition 9.8 says:

- For every type expression A and type variable X, the map $f : t \mapsto \overline{\mathcal{I}}(A, \xi[t/X])$ is non-expansive.
- In particular, if A is proper in X, then the map f is contractive.

Although we can also justify the rules ($\simeq$-fix) and ($\simeq$-uniq) for the equality $\simeq$ and ($\preceq$-μ) for the subtyping relation $\preceq$ by regarding $\overline{\mathcal{I}}(\mu X.A, \xi)$ as the uniq fixed point of the contractive map $f : t \mapsto \overline{\mathcal{I}}(A, \xi[t/X])$ with the help of the Banach fixed-point theorem, we take a more direct approach to show the soundness of such rules.

Lemma 9.10. *Let $\mathcal{I} = <\mathcal{T}, \sqsubseteq, \theta, \boxed{\bullet}, \boxed{\to}>$ be an interpretation of types, and ξ a type environment. If $A \simeq B$, then $\overline{\mathcal{I}}(A, \xi) = \overline{\mathcal{I}}(B, \xi)$.*

Proof. By induction on the derivation of $A \simeq B$, and by cases of the last rule in the derivation. In this proof we need only the conditions (i1) through (t2) of Definition 9.4.

Cases: ($\simeq$*-reflex*), ($\simeq$*-symm*) *and* ($\simeq$*-trans*). Obvious because $=$ is an equivalence relation.

Case: ($\simeq$-$\bullet$). In this case, there exist some A' and B' such that $A = \bullet A'$, $B = \bullet B'$ and $A' \simeq B'$. We have $\overline{\mathcal{I}}(A', \xi) = \overline{\mathcal{I}}(B', \xi)$ by the induction hypothesis. Therefore, $\overline{\mathcal{I}}(\bullet A', \xi) = \boxed{\bullet}(\overline{\mathcal{I}}(A', \xi)) = \boxed{\bullet}(\overline{\mathcal{I}}(B', \xi)) = \overline{\mathcal{I}}(\bullet B', \xi)$ by Definition 9.6.

Case: ($\simeq$-$\to$). Similar to the previous case.

Case: ($\simeq$-$\to\top$). Trivial from Definition 9.6.

Case: ($\simeq$*-fix*). For some X and A', $A = \mu X.A'$, $B = A'[\mu X.A'/X]$. Therefore, $\overline{\mathcal{I}}(\mu X.A', \xi) = \overline{\mathcal{I}}(A'[\mu X.A'/X], \xi) = \overline{\mathcal{I}}(B, \xi)$ by Definition 9.6.

Case: ($\simeq$*-uniq*). There exist some X and C such that $B = \mu X.C$, $A \simeq C[A/X]$ and C is proper in X. By the induction hypothesis,

$$\overline{\mathcal{I}}(A, \xi') = \overline{\mathcal{I}}(C[A/X], \xi') \text{ for every } \xi'. \tag{20}$$

We show $\overline{\mathcal{I}}(A, \xi) =_{\leq k} \overline{\mathcal{I}}(\mu X.C, \xi)$ for every k by induction on k. We have

$$\overline{\mathcal{I}}(A, \xi) =_{<k} \overline{\mathcal{I}}(\mu X.C, \xi) \tag{21}$$

by the induction hypotheses on k. Therefore,

$$\begin{array}{rcll} \overline{\mathcal{I}}(A,\,\xi) & = & \overline{\mathcal{I}}(C[A/X],\,\xi) & \text{(by (20))} \\ & = & \overline{\mathcal{I}}(C,\,\xi[\overline{\mathcal{I}}(A,\,\xi)/X] & \text{(by Proposition 9.7.2)} \\ & =_{\le k} & \overline{\mathcal{I}}(C,\,\xi[\overline{\mathcal{I}}(\mu X.C,\,\xi)/X]) & \text{(by (21) and Proposition 9.8)} \\ & = & \overline{\mathcal{I}}(C[\mu X.C/X],\,\xi) & \text{(by Proposition 9.7.2)} \\ & = & \overline{\mathcal{I}}(\mu X.C,\,\xi) & \text{(by Definition 9.6)} \end{array}$$

□

Proposition 9.11. *Let $\mathcal{I} = <\mathcal{T},\,\sqsubseteq,\,\theta,\,\boxed{\bullet},\,\boxed{\to}>$ be an interpretation of types, and ξ a type environment.*

1. $\overline{\mathcal{I}}(\top,\,\xi) = \top_{\mathcal{A}}$.
2. $\overline{\mathcal{I}}(A \to B,\,\xi) = \overline{\mathcal{I}}(A,\,\xi)\,\boxed{\to}\,\overline{\mathcal{I}}(B,\,\xi)$.

Proof. For the part 1, $\overline{\mathcal{I}}(\top,\,\xi) = \overline{\mathcal{I}}(\bullet X,\,\xi[\overline{\mathcal{I}}(\top,\,\xi)/X]) = \boxed{\bullet}(\overline{\mathcal{I}}(X,\,\xi[\overline{\mathcal{I}}(\top,\,\xi)/X])) = \boxed{\bullet}(\overline{\mathcal{I}}(\top,\,\xi))$ by Definition 9.6. Therefore, for every k, $\overline{\mathcal{I}}(\top,\,\xi) =_{<k} \top_{\mathcal{A}}$ implies $\overline{\mathcal{I}}(\top,\,\xi) =_{\le k} \top_{\mathcal{A}}$, by (i4) and (t1) of Definition 9.4. We now get $\overline{\mathcal{I}}(\top,\,\xi)_k = \top_{\mathcal{T}}$ by induction on k. As for the part 2, if B is not a $\top$-variant, trivial from Definition 9.6. Otherwise, $\overline{\mathcal{I}}(B,\,\xi) = \top_{\mathcal{A}}$ by the part 1 and Lemma 9.10; and therefore, $\overline{\mathcal{I}}(A,\,\xi)\,\boxed{\to}\,\overline{\mathcal{I}}(B,\,\xi) = \overline{\mathcal{I}}(A,\,\xi)\,\boxed{\to}\,\top_{\mathcal{A}} = \top_{\mathcal{A}}\,\boxed{\to}\,\top_{\mathcal{A}}$ by Definition 9.4 (t2). □

Definition 9.12. Let $\mathcal{I} = <\mathcal{T},\,\sqsubseteq,\,\theta,\,\boxed{\bullet},\,\boxed{\to}>$ be an interpretation of types, ξ a type environment, and γ a subtyping assumption. We write $\xi \models \gamma$ if and only if $\xi(X) \sqsubseteq \xi(Y)$ for every $\{X \preceq Y\} \subset \gamma$.

Lemma 9.13. *Let $\mathcal{I} = <\mathcal{T},\,\sqsubseteq,\,\theta,\,\boxed{\bullet},\,\boxed{\to}>$ be an interpretation of types, and ξ a type environment. If $\gamma \vdash A \preceq B$ is derivable and $\xi \models \gamma$, then $\overline{\mathcal{I}}(A,\,\xi) \sqsubseteq \overline{\mathcal{I}}(B,\,\xi)$.*

Proof. By induction on the derivation of $\gamma \vdash A \preceq B$, and by cases of the last subtyping rule applied in the derivation.

Case: ($\preceq$*-assump*). In this case, $A = X$ and $B = Y$ for some X and Y such that $\{X \preceq Y\} \subset \gamma$. We get $\xi(X) \sqsubseteq \xi(Y)$ from $\xi \models \gamma$.

Case: ($\preceq$-$\top$). Obvious from Proposition 9.11.1.

Case: ($\preceq$*-reflex*). Obvious from Lemma 9.10.

Case: ($\preceq$*-trans*). For some C, γ_1 and γ_2 such that $\gamma = \gamma_1 \cup \gamma_2$, the derivation ends with:

$$\dfrac{\begin{array}{cc} \vdots & \vdots \\ \gamma_1 \vdash A \preceq C & \gamma_2 \vdash C \preceq B \end{array}}{\gamma \vdash A \preceq B}\ (\preceq\text{-trans})$$

Since $\gamma = \gamma_1 \cup \gamma_2$, we get $\xi \models \gamma_1$ and $\xi \models \gamma_2$ from $\xi \models \gamma$. Therefore, $\overline{\mathcal{I}}(A,\,\xi) \sqsubseteq \overline{\mathcal{I}}(C,\,\xi) \sqsubseteq \overline{\mathcal{I}}(B,\,\xi)$ by the induction hypothesis.

Case: ($\preceq$-$\bullet$). For some A' and B' such that $A = \bullet A'$ and $B = \bullet B'$, the derivation ends with:

$$\dfrac{\begin{array}{c} \vdots \\ \gamma \vdash A' \preceq B' \end{array}}{\gamma \vdash \bullet A' \preceq \bullet B'}\ (\preceq\text{-}\bullet)$$

We get $\overline{\mathcal{I}}(A',\,\xi) \sqsubseteq \overline{\mathcal{I}}(B',\,\xi)$ by the induction hypothesis. Therefore, $\overline{\mathcal{I}}(\bullet A',\,\xi) = \boxed{\bullet}(\overline{\mathcal{I}}(A',\,\xi)) \sqsubseteq \boxed{\bullet}(\overline{\mathcal{I}}(B',\,\xi)) = \overline{\mathcal{I}}(\bullet B',\,\xi)$ by the condition (s1) of Definition 9.4 and Definition 9.6.

Case: ($\preceq$-$\rightarrow$). For some A_1, A_2, B_1, B_2, γ_1 and γ_2 such that $A = A_1 \rightarrow A_2$, $B = B_1 \rightarrow B_2$ and $\gamma = \gamma_1 \cup \gamma_2$, the derivation ends with:

$$\dfrac{\begin{array}{cc}\vdots & \vdots \\ \gamma_1 \vdash B_1 \preceq A_1 & \gamma_2 \vdash A_2 \preceq B_2\end{array}}{\gamma_1 \cup \gamma_2 \vdash A_1 \rightarrow A_2 \preceq B_1 \rightarrow B_2}\ (\preceq\text{-}\rightarrow)$$

Similarly to the previous case, we get $\overline{\mathcal{I}}(B_1, \xi) \sqsubseteq \overline{\mathcal{I}}(A_1, \xi)$ and $\overline{\mathcal{I}}(A_2, \xi) \sqsubseteq \overline{\mathcal{I}}(B_2, \xi)$ by the induction hypothesis. Therefore, $\overline{\mathcal{I}}(A_1 \rightarrow A_2, \xi) = \overline{\mathcal{I}}(A_1, \xi) \boxed{\rightarrow} \overline{\mathcal{I}}(A_2, \xi) \sqsubseteq \overline{\mathcal{I}}(B_1, \xi) \boxed{\rightarrow} \overline{\mathcal{I}}(B_2, \xi) = \overline{\mathcal{I}}(B_1 \rightarrow B_2, \xi)$ by the condition (s2) of Definition 9.4 and Definition 9.6.

Case: ($\preceq$-*approx*). $B = \bullet A$ in this case. We get $\overline{\mathcal{I}}(A, \xi) \sqsubseteq \boxed{\bullet}(\overline{\mathcal{I}}(A, \xi)) = \overline{\mathcal{I}}(\bullet A, \xi)$ by the condition (s3) of Definition 9.4 and Definition 9.6.

Case: ($\preceq$-$\rightarrow\bullet$). In this case, there exist some A_1 and A_2 such that $A = A_1 \rightarrow A_2$ and $B = \bullet A_1 \rightarrow \bullet A_2$. We get $\overline{\mathcal{I}}(A_1 \rightarrow A_2, \xi) = \overline{\mathcal{I}}(A_1, \xi) \boxed{\rightarrow} \overline{\mathcal{I}}(A_2, \xi) \sqsubseteq \boxed{\bullet}(\overline{\mathcal{I}}(A_1, \xi)) \boxed{\rightarrow} \boxed{\bullet}(\overline{\mathcal{I}}(A_2, \xi)) = \overline{\mathcal{I}}(\bullet A_1 \rightarrow \bullet A_2, \xi)$ by the condition (s4) of Definition 9.4 and Definition 9.6.

Case: ($\preceq$-$\bullet\rightarrow$). In this case, there exist some A_1 and A_2 such that $A = \bullet A_1 \rightarrow \bullet A_2$ and $B = \bullet(A_1 \rightarrow A_2)$. We get $\overline{\mathcal{I}}(\bullet A_1 \rightarrow \bullet A_2, \xi) = \boxed{\bullet}(\overline{\mathcal{I}}(A_1, \xi)) \boxed{\rightarrow} \boxed{\bullet}(\overline{\mathcal{I}}(A_2, \xi)) \sqsubseteq \boxed{\bullet}(\overline{\mathcal{I}}(A_1, \xi) \boxed{\rightarrow} \overline{\mathcal{I}}(A_2, \xi)) = \overline{\mathcal{I}}(\bullet(A_1 \rightarrow A_2), \xi)$ by the condition (s5) of Definition 9.4 and Definition 9.6.

Case: ($\preceq$-μ). For some X, Y, A' and B' such that $A = \mu X.A'$ and $B = \mu Y.B'$, the derivation ends with:

$$\dfrac{\begin{array}{c}\vdots \\ \gamma \cup \{X \preceq Y\} \vdash A' \preceq B'\end{array}}{\gamma \vdash \mu X.A' \preceq \mu Y.B'}\ (\preceq\text{-}\mu)$$

where $X \notin FTV(\gamma) \cup FTV(B')$, $Y \notin FTV(\gamma) \cup FTV(A')$, and A' and B' are proper in X and Y, respectively. We show $\overline{\mathcal{I}}(\mu X.A', \xi) \sqsubseteq_{\leq k} \overline{\mathcal{I}}(\mu Y.B', \xi)$ for every k by induction on k. Let $\xi' = \xi[\overline{\mathcal{I}}(A, \xi)|_{<k}/X, \overline{\mathcal{I}}(B, \xi)|_{<k}/Y]$. Since $\xi' \models \gamma \cup \{X \preceq Y\}$ by the induction hypothesis on k, we get

$$\overline{\mathcal{I}}(A', \xi') \sqsubseteq \overline{\mathcal{I}}(B', \xi') \tag{22}$$

by the induction hypothesis on the derivation. On the other hand,

$$\begin{array}{rcll}
\overline{\mathcal{I}}(\mu X.A', \xi) & = & \overline{\mathcal{I}}(A'[\mu X.A'/X], \xi) & \text{(by Definition 9.6)} \\
& = & \overline{\mathcal{I}}(A', \xi[\overline{\mathcal{I}}(\mu X.A', \xi)/X]) & \text{(by Lemma 9.7.2))} \\
& =_{\leq k} & \overline{\mathcal{I}}(A', \xi[\overline{\mathcal{I}}(\mu X.A', \xi)|_{<k}/X]) & \text{(by Lemma 9.8))} \\
& = & \overline{\mathcal{I}}(A', \xi') & \text{(by Lemma 9.7.1 since } Y \notin FTV(A'))
\end{array}$$

Similarly, $\overline{\mathcal{I}}(\mu Y.B', \xi) =_{\leq k} \overline{\mathcal{I}}(B', \xi')$. We therefore get $\overline{\mathcal{I}}(\mu X.A', \xi) \sqsubseteq_{\leq k} \overline{\mathcal{I}}(\mu Y.B', \xi)$ by (22). □

10 A realizability interpretation

In this section, we define a realizability interpretation of $\lambda\bullet\mu$, and show soundness of $\lambda\bullet\mu$ with respect to the interpretation.

Definition 10.1 (Realizability interpretations of $\lambda\bullet\mu$). A *realizability interpretation* of $\lambda\bullet\mu$ is a tuple $<\mathcal{V}, \cdot, \sigma, [\![\ ]\!]^{\mathcal{V}}, \mathcal{K}, \theta>$ such that:

1. $<\mathcal{V}, \cdot, \sigma, [\![\ ]\!]^{\mathcal{V}}>$ is a β-model of **Exp**.
2. $\mathcal{K} \subset \mathcal{V}$.
3. $u \cdot v \in \mathcal{K}$ for every $u \in \mathcal{K}$ and $v \in \mathcal{V}$.
4. $\theta : \textbf{TConst} \rightarrow \mathcal{A}(\{ S \mid \mathcal{K} \subset S \subset \mathcal{V} \}, \subset)$.

5. $\sigma(c) \in \theta(\tau(c))_k$ for every c and k.

Definition 10.2. Let $\mathcal{I} = <\mathcal{V},\ \cdot,\ \sigma,\ [\![\ ]\!]^{\mathcal{V}},\ \mathcal{K},\ \theta>$ be a realizability interpretation of $\lambda\bullet\mu$. We define an interpretation of types $\mathcal{I}^{\mathbf{r}} = <\mathcal{T},\ \sqsubseteq,\ \theta,\ \boxed{\bullet},\ \boxed{\rightarrow}>$, as follows:

1. $\mathcal{T} = \{\ T \mid \mathcal{K} \subset T \subset \mathcal{V}\ \}$
2. $T \sqsubseteq S$ if and only if $T \subset S$.
3. $\boxed{\bullet}(t)_k = \{\ u \in \mathcal{V} \mid u \in t_l \text{ for every } l < k\ \}$
4. $(t \boxed{\rightarrow} s)_k = \left\{ u \;\middle|\; \begin{array}{l} 1.\ u \in (t \boxed{\rightarrow} s)_l \text{ for every } l < k. \\ 2.\ u \cdot v \in s_k \text{ for every } v \in t_k. \\ 3.\ u \in \mathcal{K} \text{ or } u = [\![\lambda x.\ M]\!]^{\mathcal{V}}_{\rho} \text{ for some } x, \rho \text{ and } M. \end{array} \right\}$

We can easily check that $\mathcal{I}^{\mathbf{r}}$ is an interpretation of types. We often write $\mathcal{I}(A)^{\xi}$ and $\mathcal{I}(A)^{\xi}_k$ instead of $\overline{\mathcal{I}^{\mathbf{r}}}(A,\ \xi)$ and $\overline{\mathcal{I}^{\mathbf{r}}}(A,\ \xi)_k$, respectively. It should be noted that $\mathcal{I}(A)^{\xi}$ satisfies the following equations:

$$\begin{aligned}
\mathcal{I}(P)^{\xi}_k &= \theta(P)_k \\
\mathcal{I}(X)^{\xi}_k &= \xi(X)_k \\
\mathcal{I}(\bullet A)^{\xi}_k &= \{\ u \mid u \in \mathcal{I}(A)^{\xi}_l \text{ for every } l < k\ \} \\
\mathcal{I}(A \to B)^{\xi}_k &= \left\{ u \;\middle|\; \begin{array}{l} 1.\ u \in \mathcal{I}(A \to B)^{\xi}_l \text{ for every } l < k. \\ 2.\ u \cdot v \in \mathcal{I}(B)^{\xi}_k \text{ for every } v \in \mathcal{I}(A)^{\xi}_k. \\ 3.\ u \in \mathcal{K} \text{ or } u = [\![\lambda x.\ M]\!]^{\mathcal{V}}_{\rho} \text{ for some } x, \rho \text{ and } M. \end{array} \right\} \\
\mathcal{I}(\mu X.A)^{\xi}_k &= \mathcal{I}(A[\mu X.A/X])^{\xi}_k
\end{aligned}$$

The definition of $\mathcal{I}(\bullet A)$ says that:

- $\mathcal{I}(\bullet A)^{\xi}_0 = \mathcal{V}$, and
- $\mathcal{I}(\bullet A)^{\xi}_{k+1} = \mathcal{I}(A)^{\xi}_k$.

The set $\mathcal{K}$ takes a rather technical role (cf. [21]) in this semantics, and is only used to show head normalizability of λ-terms of certain types in the proof of Theorem 12.21. It can usually be considered an empty set. The third condition for $u \in \mathcal{I}(A \to B)^{\xi}_k$ implies that we distinguish $\lambda x.\ Mx$ from M unless $M \underset{\beta}{=} \lambda y.\ N$ for some y and N. Note that $\mathcal{I}(\bullet(A \to B))^{\xi}_0 = \mathcal{V}$ whereas $\mathcal{I}(\bullet A \to \bullet B)^{\xi}_0 \neq \mathcal{V}$. Thus, $\bullet(A \to B) \simeq \bullet A \to \bullet B$ is not valid in this interpretation. It should be noted that if we had $\mathcal{I}(\bullet(A \to B))^{\xi}_0 = \mathcal{I}(\bullet A \to \bullet B)^{\xi}_0$, then it would be also the case that $\mathcal{I}(\bullet(A \to B))^{\xi}_k = \mathcal{I}(\bullet A \to \bullet B)^{\xi}_k$ for every k. We can also consider a variant system of $\lambda\bullet\mu$ with this equality, where we have to drop the third condition from $u \in \mathcal{I}(A \to B)^{\xi}_k$ to get its soundness. However, we omit the details from this paper.

The typing system $\lambda\bullet\mu$ is sound with respect to this semantics.

Lemma 10.3. *1. If $A \simeq B$, then $\mathcal{I}(A)^{\xi} = \mathcal{I}(B)^{\xi}$.*
2. If $A \preceq B$, then $\mathcal{I}(A)^{\xi} \sqsubseteq \mathcal{I}(B)^{\xi}$.

Proof. Straightforward from Lemmas 9.10 and 9.13, respectively, since $\mathcal{I}^{\mathbf{r}}$ is an interpretation of type. Note that $\mathcal{I}(A)^{\xi} = \overline{\mathcal{I}^{\mathbf{r}}}(A,\ \xi)$. □

Theorem 10.4 (Soundness). *Let the tuple $<\mathcal{V},\ \cdot,\ \sigma,\ [\![\ ]\!]^{\mathcal{V}},\ \mathcal{K},\ \theta>$ be a realizability interpretation of $\lambda\bullet\mu$, and ξ a type environment. If $\{x_1 : A_1,\ \ldots,\ x_n : A_n\} \vdash M : B$ is derivable in $\lambda\bullet\mu$, then $[\![M]\!]^{\mathcal{V}}_{\rho} \in \mathcal{I}(B)^{\xi}_k$ for every k, ξ and ρ provided $\rho(x_i) \in \mathcal{I}(A_i)^{\xi}_k$ 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 Lemma 10.3 for the case of $(\preceq)$. Prove it by induction on k in the case of $(\to\mathrm{I})$.

Let $\Gamma = \{x_1 : A_1,\ \ldots,\ x_n : A_n\}$, and suppose that $\Gamma \vdash M : B$ is derivable, and $\rho(x_i) \in \mathcal{I}(A_i)^{\xi}_k$ ($i = 1,\ 2,\ \ldots, n$). In the sequel, let $[\![L]\!]$ and $\mathcal{I}(A)_k$ be abbreviations for $[\![L]\!]^{\mathcal{V}}_{\rho}$ and $\mathcal{I}(A)^{\xi}_k$, respectively.

Case: (var). $M = x_j$ and $B = A_j$ for some j ($1 \leq j \leq n$). Therefore, by assumption, $[\![M]\!] = [\![x_j]\!] = \rho(x_j) \in \mathcal{I}(A_j)_k = \mathcal{I}(B)_k$.

Case: (*const*). $M = c$ and $B = \tau(c)$ for some c. Since $<\mathcal{V}, \cdot, \sigma, [\![\ ]\!]^{\mathcal{V}}, \mathcal{K}, \theta>$ is a realizability interpretation of $\lambda{\bullet}\mu$, $[\![M]\!] = [\![c]\!] = \sigma(c) \in \theta(\tau(c))_k = \mathcal{I}(\tau(c))_k = \mathcal{I}(B)_k$.

Case: ($\top$). Obvious since $\mathcal{I}(\top)_k = \mathcal{V}$ for every k by Proposition 9.11.1.

Case: ($\bullet$). The derivation ends with:

$$\dfrac{\begin{matrix}\vdots\\ \{x_1 : \bullet A_1,\, x_2 : \bullet A_2,\, \ldots,\, x_n : \bullet A_n\} \;\vdash\; M : \bullet B\end{matrix}}{\{x_1 : A_1,\, x_2 : A_2,\, \ldots,\, x_n : A_n\} \;\vdash\; M : B}\ (\bullet)$$

Since $\rho(x_i) \in \mathcal{I}(A_i)_k$, we have $\rho(x_i) \in \mathcal{I}(\bullet A_i)_{k+1}$ for every i. Therefore, by the induction hypothesis, we get $[\![M]\!] \in \mathcal{I}(\bullet B)_{k+1}$, that is, $[\![M]\!] \in \mathcal{I}(B)_k$.

Case: ($\preceq$). For some B', the derivation ends with:

$$\dfrac{\begin{matrix}\vdots\\ \Gamma \;\vdash\; M : B'\end{matrix} \qquad \begin{matrix}\vdots\\ \vdash\; B' \preceq B\end{matrix}}{\Gamma \;\vdash\; M : B}\ (\preceq)$$

We get $[\![M]\!] \in \mathcal{I}(B)_k$ by Proposition 10.3 because $[\![M]\!] \in \mathcal{I}(B')_k$ by the induction hypothesis.

Case: ($\to$*I*). For some y, L, B_1 and B_2 such that $y \neq x_i$ ($i = 1, 2, \ldots, n$), $M = \lambda y.\, L$ and $B = B_1 \to B_2$, the derivation ends with:

$$\dfrac{\begin{matrix}\vdots\\ \Gamma \cup \{y : B_1\} \;\vdash\; L : B_2\end{matrix}}{\Gamma \;\vdash\; \lambda y.\, L : B_1 \to B_2}\ (\to\mathrm{I})$$

Since $\rho(x_i) \in \mathcal{I}(A_i)^{\xi}_k$ for every i, we also have $\rho(x_i) \in \mathcal{I}(A_i)^{\xi}_m$ for every i and $m \leq k$. Therefore, by the induction hypothesis on the derivation, we have:

$$\text{for every } m \leq k,\ [\![L]\!]^{\mathcal{V}}_{\rho} \in \mathcal{I}(B_2)_m \text{ provided } \rho(y) \in \mathcal{I}(B_1)_m. \tag{23}$$

We show $[\![\lambda y.\, L]\!] \in \mathcal{I}(B_1 \to B_2)_m$ for every $m \leq k$ by accumulative induction on m. The first condition, $[\![\lambda y.\, L]\!] \in \mathcal{I}(B_1 \to B_2)_l$ for every $l < m$, is trivial from the induction hypothesis on m, and the second one is also established because $[\![\lambda y.\, L]\!]^{\mathcal{V}}_{\rho} \cdot u = [\![L]\!]^{\mathcal{V}}_{\rho[u/y]} \in \mathcal{I}(B_2)_m$ by (23), whenever $u \in \mathcal{I}(B_1)_m$. The last condition is trivial.

Case: ($\to$*E*). For some n, M_1, M_2, B' and C such that $M = M_1 M_2$ and $B = \bullet^n B'$, the derivation ends with:

$$\dfrac{\begin{matrix}\vdots\\ \Gamma \;\vdash\; M_1 : \bullet^n(C \to B')\end{matrix} \qquad \begin{matrix}\vdots\\ \Gamma \;\vdash\; M_2 : \bullet^n C\end{matrix}}{\Gamma \;\vdash\; M_1 M_2 : \bullet^n B'}\ (\to\mathrm{E})$$

We assume that $k \geq n$ since $[\![M]\!] \in \mathcal{I}(B)_k$ is trivial if $k < n$. By the induction hypothesis, we get $[\![M_1]\!] \in \mathcal{I}(\bullet^n(C \to B'))_k = \mathcal{I}(C \to B')_{k-n}$, and $[\![M_2]\!] \in \mathcal{I}(\bullet^n C)_k = \mathcal{I}(C)_{k-n}$. Therefore, $[\![M_1 M_2]\!] = [\![M_1]\!] \cdot [\![M_2]\!] \in \mathcal{I}(B')_{k-n} = \mathcal{I}(\bullet^n B')_k$ by the definition of $\mathcal{I}(C \to B')$. □

11 Soundness of $\boldsymbol{\lambda{\bullet}\mu}$ with respect to the models of $\boldsymbol{\lambda\mu}$

In this section, we show that the typing system $\lambda{\bullet}\mu$ is also sound with respect to a certain class of models of $\lambda\mu$ if we ignore the $\bullet$-operators.

Definition 11.1. Let A be a type expression of $\lambda{\bullet}\mu$. We define $\mathrm{er}(A)$ as the type expression (of $\lambda\mu$) obtained from A by erasing all occurrences of $\bullet$ from A. We similarly define $\mathrm{er}(\Gamma)$ for typing contexts.

Lemma 11.2. *Let A and B be type expressions of $\lambda\bullet\mu$. Then,* $\mathrm{er}(A[B/X]) = \mathrm{er}(A)[\mathrm{er}(B)/X]$.

Proof. By induction on $h(A)$, and by cases of the form of A. □

Proposition 11.3. *Let $\mu X.A$ be a type expression of $\lambda\bullet\mu$. If* $\mathrm{er}(A)$ *is not $\lambda\mu$-proper in X, then $\mu X.A \simeq \top$.*

Proof. It suffices to show that $A \simeq \bullet^n X$ for some n. By straightforward induction on $h(A)$. □

Proposition 11.4. *Let A be a type expression of $\lambda\bullet\mu$. If $A \simeq \top$, then* $\mathrm{er}(A) \simeq_{\lambda\mu} \mu X.X$.

Proof. If $A \simeq \top$, i.e., A is a $\top$-variant, then $\mathrm{er}(A) = \mu X_1.\mu X_2.\ldots\mu X_n.X_i$ for some n, X_1, X_2, …, X_n and i such that $1 \leq i \leq n$; and therefore, $\mathrm{er}(A) \simeq_{\lambda\mu} \mu X.X$ by ($\simeq_{\lambda\mu}$-fix) and Proposition 3.6. □

Definition 11.5. A realizability model $<\mathcal{V}, \cdot, \sigma, [\![\ ]\!]^{\mathcal{V}}, \mathcal{T}, \delta, [\![\ ]\!]^{\mathcal{T}}>$ of $\lambda\mu$ is *standard* if and only if it satisfies the following:

1. $[\![\mu X.X]\!]^{\mathcal{T}}_{\eta} = \mathcal{V}$.
2. If C is $\lambda\mu$-proper in X, and $[\![A]\!]^{\mathcal{T}}_{\eta} = [\![C[A/X]]\!]^{\mathcal{T}}_{\eta}$ for every η, then $[\![A]\!]^{\mathcal{T}}_{\eta} = [\![\mu X.C]\!]^{\mathcal{T}}_{\eta}$ for every η.
3. If A and B are $\lambda\mu$-proper in X and Y, respectively, and if $X \notin FTV(B)$, $Y \notin FTV(A)$, and $[\![A]\!]^{\mathcal{T}}_{\eta} \subset [\![B]\!]^{\mathcal{T}}_{\eta}$ for every η, then $[\![\mu X.A]\!]^{\mathcal{T}}_{\eta} \subset [\![\mu Y.B]\!]^{\mathcal{T}}_{\eta}$ for every η.

Lemma 11.6. *Let $<\mathcal{V}, \cdot, \sigma, [\![\ ]\!]^{\mathcal{V}}, \mathcal{T}, \delta, [\![\ ]\!]^{\mathcal{T}}>$ be a standard realizability model of $\lambda\mu$. If $A \simeq B$, then* $[\![\mathrm{er}(A)]\!]^{\mathcal{T}}_{\eta} = [\![\mathrm{er}(B)]\!]^{\mathcal{T}}_{\eta}$.

Proof. By induction on the derivation of $A \simeq B$, and by cases of the last rule applied in the derivation. The condition $[\![\mu X.X]\!]^{\mathcal{T}}_{\eta} = \mathcal{V}$ is required to validate the ($\simeq$-$\rightarrow\top$)-rule. If the last rule is ($\simeq$-fix), then $A = \mu X.A'$ and $B = A'[\mu X.A'/X]$ for some X and A'.

$$\begin{array}{rcll} \mathrm{er}(A) &=& \mathrm{er}(\mu X.A') & \\ &=& \mu X.\,\mathrm{er}(A') & \\ &\simeq_{\lambda\mu}& \mathrm{er}(A')[\mu X.\,\mathrm{er}(A')/X] & \text{(by}\quad (\simeq_{\lambda\mu}\text{-fix))} \\ &=& \mathrm{er}(A')[\mathrm{er}(\mu X.A')/X] & \\ &=& \mathrm{er}(A'[\mu X.A'/X]) & \text{(by Proposition 11.2)} \\ &=& \mathrm{er}(B) & \end{array}$$

Therefore, $[\![\mathrm{er}(A)]\!]^{\mathcal{T}}_{\eta} = [\![\mathrm{er}(B)]\!]^{\mathcal{T}}_{\eta}$ by Definition 3.9.9. If the last rule is ($\simeq$-uniq), then $B = \mu X.C$ and $A \simeq C[A/X]$ for some X and C such that C is proper in X. For every η, we get:

$$\begin{array}{rll} [\![\mathrm{er}(A)]\!]^{\mathcal{T}}_{\eta} &= [\![\mathrm{er}(C[A/X])]\!]^{\mathcal{T}}_{\eta} & \text{(by the induction hypothesis)} \\ &= [\![\mathrm{er}(C)[\mathrm{er}(A)/X]]\!]^{\mathcal{T}}_{\eta} & \text{(by}\quad\text{Proposition}\quad 11.2). \end{array}$$

If $\mathrm{er}(C)$ is $\lambda\mu$-proper in X, then $[\![\mathrm{er}(A)]\!]^{\mathcal{T}}_{\eta} = [\![\mu X.\,\mathrm{er}(C)]\!]^{\mathcal{T}}_{\eta} = [\![\mathrm{er}(B)]\!]^{\mathcal{T}}_{\eta}$ by Definition 11.5.2. Otherwise, $A \simeq B = \mu X.C \simeq \top$ by Proposition 11.3; and therefore, $\mathrm{er}(A) \simeq_{\lambda\mu} \mathrm{er}(B)$ by Proposition 11.4. We now get $[\![\mathrm{er}(A)]\!]^{\mathcal{T}}_{\eta} = [\![\mathrm{er}(B)]\!]^{\mathcal{T}}_{\eta}$ by Definition 3.9.9. □

Lemma 11.7. *Let $<\mathcal{V}, \cdot, \sigma, [\![\ ]\!]^{\mathcal{V}}, \mathcal{T}, \delta, [\![\ ]\!]^{\mathcal{T}}>$ be a standard realizability model of $\lambda\mu$. If $A \preceq B$, then* $[\![\mathrm{er}(A)]\!]^{\mathcal{T}}_{\eta} \subset [\![\mathrm{er}(B)]\!]^{\mathcal{T}}_{\eta}$.

Proof. Prove a more general statement: if $\{X_1 \preceq Y_1, X_2 \preceq Y_2, \ldots, X_n \preceq Y_n\} \vdash A \preceq B$ is derivable, then $[\![\mathrm{er}(A[X_1/Y_1, X_2/Y_2, \ldots, X_n/Y_n])]\!]^{\mathcal{T}}_{\eta} \subset [\![\mathrm{er}(B[X_1/Y_1, X_2/Y_2, \ldots, X_n/Y_n])]\!]^{\mathcal{T}}_{\eta}$, by induction on the derivation and by cases of the last rule applied in the derivation. Use Lemma 11.6 for the case ($\preceq$-reflex). Use Propositions 3.5 and 11.2 for ($\preceq$-trans) and ($\preceq$-$\rightarrow$). Use the condition 3 of Definition 11.5 for ($\preceq$-μ). □

Theorem 11.8 (Soundness w.r.t. standard models of $\lambda\mu$). *Let $<\mathcal{V}, \cdot, \sigma, [\![\ ]\!]^{\mathcal{V}}, \mathcal{T}, \delta, [\![\ ]\!]^{\mathcal{T}}>$ be a standard realizability model of $\lambda\mu$. If $\{x_1 : A_1, x_2 : A_2, \ldots, x_n : A_n\} \vdash M : B$ is derivable in $\lambda\bullet\mu$, then $[\![M]\!]^{\mathcal{V}}_{\rho} \in [\![\mathrm{er}(B)]\!]^{\mathcal{T}}_{\eta}$ for every η and ρ provided $\rho(x_i) \in [\![\mathrm{er}(A_i)]\!]^{\mathcal{T}}_{\eta}$ ($i = 1, 2, \ldots, n$).*

Proof. By induction on the derivation of $\{x_1 : A_1, \ldots, x_n : A_n\} \vdash M : B$, and by cases of the last rule applied in the derivation. Use Lemma 11.7 if the last rule is ($\preceq$). □

12 Convergence of well-typed terms

The soundness theorem assures the convergence of well-typed λ-terms according to their types. In this section we give such results.

Definition 12.1. The term model $<\mathcal{V},\ \cdot,\ \sigma,\ [\![\ ]\!]^{\mathcal{V}}>$ of **Exp** is defined as follows:

1. $\mathcal{V} = \textbf{Exp}/_{\overline{\beta}}$
2. $M \cdot N = M\,N$
3. $\sigma(c) = c$
4. $[\![x]\!]^{\mathcal{V}}_{\rho} = \rho(x)$
5. $[\![c]\!]^{\mathcal{V}}_{\rho} = c$
6. $[\![M\,N]\!]^{\mathcal{V}}_{\rho} = [\![M]\!]^{\mathcal{V}}_{\rho}\ [\![N]\!]^{\mathcal{V}}_{\rho}$
7. $[\![\lambda x.\,M]\!]^{\mathcal{V}}_{\rho} = \lambda x.\ [\![M]\!]^{\mathcal{V}}_{\rho[x/x]}$

Proposition 12.2. *The term model $<\mathcal{V},\ \cdot,\ \sigma,\ [\![\ ]\!]^{\mathcal{V}}>$ is a β-model of* **Exp**.

Proposition 12.3. *Let $<\mathcal{V},\ \cdot,\ \sigma,\ [\![\ ]\!]^{\mathcal{V}}>$ be the term model of* **Exp**. *If $\rho(x) = x$ for every $x \in FV(M)$, then $[\![M]\!]^{\mathcal{V}}_{\rho} = M$.*

Definition 12.4. We define a set of λ-terms $\mathcal{K}_0$ as $\mathcal{K}_0 = \{\ x N_1 N_2 \ldots N_n \mid x \in \textbf{Var},\ n \geq 0$ and $N_i \in \textbf{TExp}/_{\overline{\beta}}$ for every i $(i = 1,\ 2,\ \ldots,\ n)\ \}$.

12.1 Non $\top$ types and weakly head normalizable terms

Definition 12.5 (Weak head normal forms). A λ-term M is a *weak head normal form* if and only if M is either of the following forms:

1. c
2. $x\,N_1\,N_2\ \ldots\ N_n$, where $n \geq 0$
3. $\lambda x.\,N$

We say that M *has a weak head normal form*, or *is weakly head normalizable*, if $M \xrightarrow[\beta]{*} M'$ for some weak head normal form M'.

Proposition 12.6. *1. If $M =_{\beta} c$, then $M \xrightarrow[\beta]{*} c$.*

2. If $M =_{\beta} x N_1\,N_2\ \ldots\ N_n$, then $M \xrightarrow[\beta]{*} x N'_1\,N'_2\ \ldots\ N'_n$ for some $N'_1,\ N'_2,\ \ldots,\ N'_n$.

3. If $M =_{\beta} \lambda x.\,L$, then $M \xrightarrow[\beta]{*} \lambda x.\,L'$ for some L'.

Proof. By induction on the length of the conversion $M \underset{\beta}{\leftrightarrow} M_1 \underset{\beta}{\leftrightarrow} M_2 \underset{\beta}{\leftrightarrow} \ldots \underset{\beta}{\leftrightarrow} M_n \underset{\beta}{\leftrightarrow} c\ (\lambda x.\,L,\ x N_1\,N_2\ \ldots\ N_n)$. Use Church-Rosser property of untyped lambda calculus. □

Theorem 12.7. *If $\Gamma\ \vdash\ M : A$ is derivable for some A such that $A \not\simeq \top$, then M has a weak head normal form.*

Proof. Consider the interpretation $<\mathcal{V},\ \cdot,\ \sigma,\ [\![\ ]\!]^{\mathcal{V}},\ \mathcal{K}_0,\ \theta>$ of $\lambda\bullet\mu$, where $<\mathcal{V},\ \cdot,\ \sigma,\ [\![\ ]\!]^{\mathcal{V}}>$ is the term model of $\lambda\mu$, and $\theta(P)_k = \mathcal{K}_0 \cup \{\ M \mid M =_{\beta} c$ for some $c \in$ **Const** $\}$ for every type constant P. Let $\xi(X)_k = \mathcal{K}_0$ for every k and X.

Suppose that $A \not\simeq \top$ and $\Gamma\ \vdash\ M : A$ is derivable. Since $A \simeq A^c$, $\Gamma\ \vdash\ M : A^c$ is also derivable. Fixing ρ as $\rho(x) = x$ for every x, we get $M \in \mathcal{I}(A^c)^{\xi}_k$ for every $ttenv$ and k by Theorem 10.4, because $[\![M]\!]^{\mathcal{V}}_{\rho} = M$ by Proposition 12.3, and $\rho(x) = x \in \mathcal{K}_0 \subset \mathcal{I}(\Gamma(x))^{\xi}_k$ for every $x \in Dom(\Gamma)$. We now show that M has a weak head normal form by cases of the form of A^c.

Case: $A^c = \bullet^n P$. M obviously has a weak head normal form by Proposition 12.6, because $M \in \mathcal{I}(A^c)^{\xi}_n = \mathcal{I}(P)^{\xi}_0 = \theta(P)_k = \mathcal{K}_0 \cup \{\ M \mid M =_{\beta} c$ for some $c \in$ **Const** $\}$.

Case: $A^c = \bullet^n X$. Also trivial by Proposition 12.6, because $M \in \mathcal{I}(A^c)^{\xi}_n = \mathcal{I}(X)^{\xi}_0 = \xi(X)_0 = \mathcal{K}_0$.

Case: $A^c = \bullet^n(B \rightarrow C)$. In this case, $M \in \mathcal{I}(A^c)^{\xi}_n = \mathcal{I}(B \rightarrow C)^{\xi}_0 \subset \mathcal{K}_0 \cup \{\ M \mid M =_{\beta} \lambda y.\,L\ \}$. Therefore, N has a weak head normal form. □

12.2 Tail finite types

In this subsection, we show that every term of *tail finite* types is *head normalizable*. First, we define head normalizable terms in a standard manner.

Definition 12.8 (Head normal forms). A λ-term M is a *head normal form* if and only if M is either of the following forms:

1. c
2. $\lambda x_1. \lambda x_2. \ldots \lambda x_m. y\, N_1\, N_2\, \ldots\, N_n$, where $m,\, n \geq 0$

We say that M *has a head normal form*, or *is head normalizable*, if $M \xrightarrow[\beta]{*} M'$ for some head normal form M'. We also define Böhm trees of λ-terms in the standard manner according to this definition of head normal forms, in which λ-terms without head normal forms are denoted by $\perp$.

Next we define tail finite type expressions in three different ways, i.e., semantically, syntactically, and in a somewhat intuitive way. We start with semantical definition.

Definition 12.9. We define an interpretation of types $\mathcal{I}^{\mathbf{t}} = <\mathcal{T}^{\mathbf{t}},\, \sqsubseteq^{\mathbf{t}},\, \theta^{\mathbf{t}},\, \boxdot^{\mathbf{t}},\, \boxminus\!\!\!\rightarrow^{\mathbf{t}}>$, as follows:

1. $\mathcal{T}^{\mathbf{t}} = \mathcal{N}^+$, where $\mathcal{N}^+ = \{0,\, 1,\, 2,\, \ldots,\, \infty\}$.
2. $n \sqsubseteq^{\mathbf{t}} m$ if and only if $n \leq m$
3. $\theta^{\mathbf{t}}(P)_k = 0$
4. $\boxdot^{\mathbf{t}}(t)_k = \inf_{l<k} t_l$, where $\boxdot^{\mathbf{t}}(t)_0 = \infty$
5. $(t \boxminus\!\!\!\rightarrow^{\mathbf{t}} s)_k = s_k + 1$

We can easily check that $\mathcal{I}^{\mathbf{t}}$ is an interpretation of types.

Lemma 12.10. *Suppose that* $\xi(X) \sqsubseteq^{\mathbf{t}} \xi'(X)$ *for every* $X \in$ **TVar**. *Then* $\overline{\mathcal{I}^{\mathbf{t}}}(A,\, \xi) \sqsubseteq^{\mathbf{t}} \overline{\mathcal{I}^{\mathbf{t}}}(A,\, \xi')$.

Proof. By induction on $h(A)$, and by cases of the form of A. The only interesting case is when $A = \mu X. A'$ for some X and A'. It suffices to show that $\overline{\mathcal{I}^{\mathbf{t}}}(\mu X. A',\, \xi) \sqsubseteq^{\mathbf{t}}_{\leq k} \overline{\mathcal{I}^{\mathbf{t}}}(\mu X. A',\, \xi')$ for every k. The proof proceeds by induction on k. By the induction hypothesis on k, we have

$$\overline{\mathcal{I}^{\mathbf{t}}}(\mu X. A',\, \xi) \sqsubseteq^{\mathbf{t}}_{<k} \overline{\mathcal{I}^{\mathbf{t}}}(\mu X. A',\, \xi'). \tag{24}$$

Note also that A' is proper in X.

$$\begin{array}{rcll}
\overline{\mathcal{I}^{\mathbf{t}}}(\mu X. A',\, \xi) & = & \overline{\mathcal{I}^{\mathbf{t}}}(A'[\mu X. A'/X],\, \xi) & \text{(by Definition 9.6)} \\
& = & \overline{\mathcal{I}^{\mathbf{t}}}(A',\, \xi[\overline{\mathcal{I}^{\mathbf{t}}}(\mu X. A',\, \xi)/X]) & \text{(by Definition 9.7.2)} \\
& \sqsubseteq^{\mathbf{t}}_{\leq k} & \overline{\mathcal{I}^{\mathbf{t}}}(A',\, \xi[\overline{\mathcal{I}^{\mathbf{t}}}(\mu X. A',\, \xi')/X]) & \text{(by (24) and Proposition 9.8)} \\
& \sqsubseteq^{\mathbf{t}} & \overline{\mathcal{I}^{\mathbf{t}}}(A',\, \xi'[\overline{\mathcal{I}^{\mathbf{t}}}(\mu X. A',\, \xi')/X]) & \text{(by the induction hypothesis on } h(A)) \\
& = & \overline{\mathcal{I}^{\mathbf{t}}}(A'[\mu X. A'/X],\, \xi') & \text{(by Proposition 9.7.2)} \\
& = & \overline{\mathcal{I}^{\mathbf{t}}}(\mu X. A',\, \xi') & \text{(by Definition 9.6)}
\end{array}$$

□

Definition 12.11. Let V be a set of type variables. We define $tl_V(A)$ $(\in \mathcal{N}^+)$ as:

$$tl_V(A) = \inf_k \overline{\mathcal{I}^{\mathbf{t}}}(A,\, \xi^{\mathbf{t}}_V)_k,$$

where $\xi^{\mathbf{t}}_V$ is a type environment for the interpretation $\mathcal{I}^{\mathbf{t}}$ defined as follows:

$$\xi^{\mathbf{t}}_V(X)_k = \begin{cases} \infty & (X \in V) \\ 0 & (X \notin V) \end{cases}$$

Proposition 12.12. *Let V be a set of type variables.*

$$tl_V(P) = 0, \qquad tl_V(X) = \xi^{\mathbf{t}}_V(X), \qquad tl_V(\bullet A) = tl_V(A),$$
$$tl_V(A \to B) = tl_V(B) + 1, \qquad tl_V(\mu X.A) = tl_V(A[\mu X.A/X]).$$

Note that $\infty + 1 = \infty$.

Proof. Obvious from Definitions 9.6 and 12.11, since $\mathcal{I}^{\mathbf{t}}$ is an interpretation of types. □

Proposition 12.13. *1. If $X \notin FTV(A)$, then $tl_V(A) = tl_{V\cup\{X\}}(A)$.*
2. If $A \simeq B$, then $tl_V(A) = tl_V(B)$.
3. If $A \preceq B$, then $tl_V(A) \leq tl_V(B)$.

Proof. Straightforward from Proposition 9.7.1, Lemma 9.10 and Lemma 9.13, respectively, since $\mathcal{I}^{\mathbf{t}}$ is an interpretation of type. □

Definition 12.14 (Tail finite types). A type expression A is *tail finite* if and only if $tl_{\{\}}(A) < \infty$.

Note that the notion "tail finite" has been defined in a semantical manner through Definition 12.11. We can give another definition for the notion syntactically, by which we can realize the decidability of this property. The equivalence between the two definitions will be shown by Proposition12.18.

Definition 12.15. Let V be a set of type variables. We define a subset $\mathbf{TF}^V$ of $\mathbf{TExp}$ as follows:

$$\begin{array}{rcll}
\mathbf{TF}^V & ::= & \mathbf{TConst} & \\
 & \mid & X & (X \in \mathbf{TVar} - V) \\
 & \mid & \bullet\, \mathbf{TF}^V & \\
 & \mid & \mathbf{TExp} \to \mathbf{TF}^V & \\
 & \mid & \mu X.\mathbf{TF}^{V\cup\{X\}} & (X \in \mathbf{TVar}).
\end{array}$$

We can easily check that $\mathbf{TF}^V$ is closed under α-conversion of type expressions. We denote $\mathbf{TF}^{\{\}}$ by $\mathbf{TF}$.

Proposition 12.16. *Let A be a type expression of $\lambda\mu$.*

1. *If $V \subset V'$, then $\mathbf{TF}^{V'} \subset \mathbf{TF}^V$.*
2. *If $X \notin ETV^+(A)$ and $A \in \mathbf{TF}^V$, then $A \in \mathbf{TF}^{V\cup\{X\}}$.*

Proof. By straightforward induction on the structure of A. □

Proposition 12.17. *Let A be a type expression, and V a set of type variables.*

1. *If $A \in \mathbf{TF}^{V\cup\{X\}}$, then $A[B/X] \in \mathbf{TF}^V$.*
2. *If $A \in \mathbf{TF}^V$ and $B \in \mathbf{TF}^V$, then $A[B/X] \in \mathbf{TF}^V$.*
3. *If $A[B/X] \in \mathbf{TF}^V$, then $A \in \mathbf{TF}^{V-\{X\}}$.*
4. *If $A[B/X] \in \mathbf{TF}^V$, then $A \in \mathbf{TF}^{V\cup\{X\}}$ or $B \in \mathbf{TF}^V$.*

Proof. By induction on the structure of A, and by cases of the form of A.

Case: A is a type constant. Trivial because $A[B/X] = A$. Use Proposition 12.16.1 for the parts 1 and 3, and Proposition 12.16.2 for the part 4.

Case: $A = Y$ for some Y. For the part 1, we get $X \neq Y$ from $A \in \mathbf{TF}^{V\cup\{X\}}$. Therefore, $A[B/X] = A \in \mathbf{TF}^{V\cup\{X\}} \subset \mathbf{TF}^V$ by Proposition 12.16.1. For the part 2, suppose that $A \in \mathbf{TF}^V$ and $B \in \mathbf{TF}^V$. We similarly get $A \in \mathbf{TF}^V$ if $X \neq Y$. Otherwise, i.e., if $X = Y$, then $A[B/X] = B \in \mathbf{TF}^V$. For the parts 3 and 4, suppose that $A[B/X] \in \mathbf{TF}^V$. If $X \neq Y$, then $A = A[B/X] \in \mathbf{TF}^V$; therefore, $A \in \mathbf{TF}^{V-\{X\}}$ by Proposition 12.16.1, and $A \in \mathbf{TF}^{V\cup\{X\}}$ by Proposition 12.16.2, since $X \notin FTV(A)$. Otherwise, i.e., if $X = Y$, then $A = X \in \mathbf{TF}^{V-\{X\}}$ by Definition 12.15, and $B = A[B/X] \in \mathbf{TF}^V$.

Case: $A = C \to D$ for some C and D. For the part 1, suppose that $A \in \mathbf{TF}^{V \cup \{X\}}$. Then, $D \in \mathbf{TF}^{V \cup \{X\}}$ by Definition 12.15. Therefore, we get $D[B/X] \in \mathbf{TF}^V$ by the induction hypothesis, and then, $A[B/X] = C[B/X] \to D[B/X] \in \mathbf{TF}^V$ by Definition 12.15. Similar for the parts 2, 3 and 4.

Case: $A = \mu Y.C$ for some Y and C. We can assume that $Y \notin FTV(B) \cup \{X\}$. For the part 1, we get $C \in \mathbf{TF}^{V \cup \{X, Y\}}$ from $A \in \mathbf{TF}^{V \cup \{X\}}$ by Definition 12.15. Therefore, $C[B/X] \in \mathbf{TF}^{V \cup \{X\}}$ by the induction hypothesis. We now get $A[B/X] = (\mu Y.C)[B/X] = \mu Y.C[B/X] \in \mathbf{TF}^V$ by Definition 12.15. Similar for the parts 2, 3 and 4. □

A type expression A is tail finite if and only if $A \in \mathbf{TF}$. This is shown by proving the following proposition.

Proposition 12.18. *Let V be a set of type variables. $tl_V(A) < \infty$ if and only if $A \in \mathbf{TF}^V$.*

Proof. By induction on $h(A)$, and by cases of the from of A. The only interesting case is when $A = \mu X.A'$ for some X and A'. First, we show the "if" part. If $\mu X.A' \in \mathbf{TF}^V$, then $A' \in \mathbf{TF}^{V \cup \{X\}}$ by Definition 12.15. Therefore,

$$\begin{aligned}
tl_V(\mu X.A') &= \inf_k \overline{\mathcal{I}^{\mathbf{t}}}(\mu X.A',\, \xi^{\mathbf{t}}_V)_k \\
&= \inf_k \overline{\mathcal{I}^{\mathbf{t}}}(A'[\mu X.A'/X],\, \xi^{\mathbf{t}}_V)_k && \text{(by Definition 9.6)} \\
&= \inf_k \overline{\mathcal{I}^{\mathbf{t}}}(A',\, \xi^{\mathbf{t}}_V[\overline{\mathcal{I}^{\mathbf{t}}}(\mu X.A',\, \xi^{\mathbf{t}}_V)/X])_k && \text{(by Proposition 9.7.2)} \\
&\le \inf_k \overline{\mathcal{I}^{\mathbf{t}}}(A',\, \xi^{\mathbf{t}}_{V \cup \{X\}})_k && \text{(by Lemma 12.10)} \\
&= tl_{V \cup \{X\}}(A') \\
&< \infty && \text{(by the induction hypothesis)}
\end{aligned}$$

For the "only if" part, we show the contrapositive. Suppose that $\mu X.A' \notin \mathbf{TF}^V$, i.e., $A' \notin \mathbf{TF}^{V \cup \{X\}}$ by Definition 12.15. We have $tl_{V \cup \{X\}}(A') = \infty$ by the induction hypothesis; and since $tl_{V \cup \{X\}}(A') = \inf_k \overline{\mathcal{I}^{\mathbf{t}}}(A',\, \xi^{\mathbf{t}}_{V \cup \{X\}})_k$,

$$\overline{\mathcal{I}^{\mathbf{t}}}(A',\, \xi^{\mathbf{t}}_{V \cup \{X\}})_k = \infty \quad \text{for every } k. \tag{25}$$

It suffices to show $\overline{\mathcal{I}^{\mathbf{t}}}(\mu X.A',\, \xi^{\mathbf{t}}_V)_k = \infty$ for every k. The proof proceeds by induction on k. By the induction hypothesis, $\overline{\mathcal{I}^{\mathbf{t}}}(\mu X.A',\, \xi^{\mathbf{t}}_V)_l = \infty$ for every $l < k$, i.e.:

$$\overline{\mathcal{I}^{\mathbf{t}}}(\mu X.A',\, \xi^{\mathbf{t}}_V) =_{<k} \xi^{\mathbf{t}}_{\{X\}}(X). \tag{26}$$

Note also that A' is proper in X.

$$\begin{aligned}
\overline{\mathcal{I}^{\mathbf{t}}}(\mu X.A',\, \xi^{\mathbf{t}}_V)_k &= \overline{\mathcal{I}^{\mathbf{t}}}(A'[\mu X.A'/X],\, \xi^{\mathbf{t}}_V)_k && \text{(by Definition 9.6)} \\
&= \overline{\mathcal{I}^{\mathbf{t}}}(A',\, \xi^{\mathbf{t}}_V[\overline{\mathcal{I}^{\mathbf{t}}}(\mu X.A',\, \xi^{\mathbf{t}}_V)/X])_k && \text{(by Proposition 9.7.2)} \\
&= \overline{\mathcal{I}^{\mathbf{t}}}(A',\, \xi^{\mathbf{t}}_V[\xi^{\mathbf{t}}_{\{X\}}/X])_k && \text{(by (26) and Proposition 9.8)} \\
&= \overline{\mathcal{I}^{\mathbf{t}}}(A',\, \xi^{\mathbf{t}}_{V \cup \{X\}})_k \\
&= \infty && \text{(by (25))}
\end{aligned}$$

□

We can define the notion "tail finite" in another way, by considering (possibly infinite) expansions of type expressions. Proposition 12.20 provides us the equivalence.

Proposition 12.19. *Let V be a set of type variables. If (a) $A \not\simeq \bullet^k X$, (b) $A \not\simeq \bullet^k (B \to C)$, and (c) $A \not\simeq \top$ for every k, X, B, C such that $X \in V$, then $tl_V(A) = 0$.*

Proof. By cases of the form of A^c. First, we get $A^c \ne \top$ and $A^c \ne \bullet^n (B \to C)$ for every n, B and C, from (b) and (c). If $A^c = \bullet^n P$ for some n and P, then obviously $tl_V(A) = 0$. If $A^c = \bullet^n X$ for some n and X, then also $tl_V(A) = 0$ since $X \notin V$ by (a). □

Proposition 12.20. *Let V be a set of type variables. $tl_V(A) = n$ if and only if $A \simeq \bullet^{m_1}(B_1 \to \bullet^{m_2}(B_2 \to \bullet^{m_3}(B_3 \to \ldots \to \bullet^{m_n}(B_n \to C)\ldots)))$ for some n, m_0, m_1, m_2, ..., m_n, B_1, B_2, ..., B_n and C such that*

(a) $C \not\simeq \bullet^k X$ for every k and $X \in V$,
(b) $C \not\simeq \bullet^k(D \to E)$ for every k, D and E, and
(c) $C \not\simeq \top$.

Proof. First, we show the "if" part. Suppose that $A \simeq \bullet^{m_1}(B_1 \to \bullet^{m_2}(B_2 \to \bullet^{m_3}(B_3 \to \ldots \to \bullet^{m_n}(B_n \to C)\ldots)))$ and (a) through (c) hold. With the help of Proposition 12.12,

$$\begin{aligned} tl_V(A) &= tl_V(\bullet^{m_1}(B_1 \to \bullet^{m_2}(B_2 \to \bullet^{m_3}(B_3 \to \ldots \to \bullet^{m_n}(B_n \to C)\ldots)))) \\ &= 1 + tl_V(\bullet^{m_2}(B_2 \to \bullet^{m_3}(B_3 \to \ldots \to \bullet^{m_n}(B_n \to C)\ldots))) \\ &\;\vdots \\ &= n + tl_V(C) \\ &= n \qquad\qquad \text{(by Proposition 12.19 and (a) through (c))} \end{aligned}$$

For the "only if" part, suppose that $tl_V(A) = n$; that is, $tl_V(A^c) = n$ by Proposition 12.13. The proof proceeds by induction on n, and by cases of the form of A^c. If $A^c = \bullet^k P$ for some k and P, then trivial since $tl_V(A^c) = 0$ in this case. If $A^c = \bullet^k X$ for some k and X, then straightforward because we get $tl_V(A^c) = 0$ and $X \notin V$ from $tl_V(\bullet^k X) = n < \infty$. If $A^c = \bullet^k(A_1 \to A_2)$ for some A_1 and A_2, then also straightforward from the induction hypothesis because $tl_V(A^c) = tl_V(A_2) + 1$. The case $A^c = \top$ is impossible because $tl_V(\top) = \infty$. □

Now we show the main results of this subsection, which is stated as the following theorem.

Theorem 12.21. *Let V be a set of type variables, and A a type expression such that $A \in \mathbf{TF}^V$. If $\Gamma \vdash M : A$ is derivable, then M has a head normal form.*

Proof. Considering the same realizability interpretation $<\mathcal{V}, \cdot, \sigma, [\![\;]\!]^{\mathcal{V}}, \mathcal{K}, \theta>$ of $\lambda\bullet\mu$ and ρ as in the proof of Theorem 12.7, we get $M \in \mathcal{I}(A)^{\xi}_k$ for every ξ and k by Theorem 10.4. Since ξ can be any type environment, it suffices to show that M has a head normal form whenever there exists some ξ and A such that

(a) $A \in \mathbf{TF}^V$,
(b) $\xi(X)_k = \mathcal{K}_0$ for every k and $X \notin V$, and
(c) $M \in \mathcal{I}(A)^{\xi}_k$ for every k.

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

Case: $A = P$. M obviously has a head normal form by Proposition 12.6, because $\mathcal{I}(A)^{\xi}_k = \theta(P)_k = \mathcal{K}_0 \cup \{\, M \mid M =_{\beta} c \text{ for some } c \in \mathbf{Const} \,\}$.

Case: $A = X$. In this case, $\mathcal{I}(A)^{\xi}_k = \xi(X)_k = \mathcal{K}_0$ by (a) and (b). M therefore has a head normal form by Proposition 12.6.

Case: $A = \bullet B$. In this case, $h(B) < h(A)$ and $B \in \mathbf{TF}^V$ by (a). Therefore, M has a head normal form by the induction hypothesis, because $M \in \mathcal{I}(\bullet B)^{\xi}_{k+1} = \mathcal{I}(B)^{\xi}_k$ for every k.

Case: $A = B \to C$. In this case, $h(C) < h(A)$ and $C \in \mathbf{TF}^V$ by (a). Let y be a fresh individual variable. Since $M \in \mathcal{I}(B \to C)^{\xi}_k$ and $y \in \mathcal{K}_0 \subset \mathcal{I}(B)^{\xi}_k$ for every k, we get $My \in \mathcal{I}(C)^{\xi}_k$ for every k. Therefore, My has a head normal form, say L, by the induction hypothesis. There are two possible cases: for some K, (1) $M \xrightarrow[\beta]{*} K \in \mathcal{K}_0$ and $L = Ky \in \mathcal{K}_0$, or (2) $M \xrightarrow[\beta]{*} \lambda y.\, K$ and $K \xrightarrow[\beta]{*} L$. In either case, M obviously has a head normal form.

Case: $A = \mu Y.B$. In this case, $h(B) < h(A)$ and $B \in \mathbf{TF}^{V \cup \{Y\}}$ by (a). By Definition 9.6 and Proposition 9.7, we get $\mathcal{I}(\mu Y.B)^{\xi}_k = \mathcal{I}(B[\mu Y.B/Y])^{\xi}_k = \mathcal{I}(B)^{\xi'}_k$, where $\xi' = \xi[\mathcal{I}(\mu Y.B)^{\xi}/Y]$. Note that (a′) $B \in \mathbf{TF}^{V \cup \{Y\}}$, (b′) $\xi'(X)_k = \mathcal{K}_0$ for every k and $X \notin V \cup \{Y\}$, and (c′) $M \in \mathcal{I}(B)^{\xi'}_k$ for every k. Therefore, M has a head normal form by the induction hypothesis. □

12.3 Positively and negatively finite types

Definition 12.22 (Maximal λ-terms). A λ-term M is *maximal* if and only if the Böhm tree of M has no occurrence of $\perp$, which represents non-head normalizable λ-terms.

Note that the maximality of λ-terms is closed under $\underset{\beta}{=}$.

Definition 12.23 (Positively and negatively finite types). 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 $X \notin ETV^{-}(B)$ ($X \notin ETV^{+}(B)$).

Note that every positively finite type expression is tail finite.

Definition 12.24. We define subsets **PF** and **NF** of **TExp** as follows:

$$\begin{array}{rcll}
\mathbf{PF} & ::= & \mathbf{TConst} \mid \mathbf{TVar} & \\
& \mid & \bullet\,\mathbf{PF} \mid \mathbf{NF} \to \mathbf{PF} & \\
& \mid & \mu X.A & (\mu X.A \in \mathbf{TF},\ A \in \mathbf{PF}, \text{ and (a) } X \notin ETV^{-}(A) \text{ or (b) } A \in \mathbf{NF}).
\end{array}$$

$$\begin{array}{rcll}
\mathbf{NF} & ::= & \mathbf{TConst} \mid \mathbf{TVar} & \\
& \mid & \bullet\,\mathbf{NF} \mid \mathbf{PF} \to \mathbf{NF} & \\
& \mid & A \to B & (A,\ B \in \mathbf{TExp}, \text{ and } B \text{ is a } \top\text{-variant}) \\
& \mid & \mu X.A & (\mu X.A \in \mathbf{TExp},\ A \in \mathbf{NF}, \text{ and (a) } X \notin ETV^{-}(A) \text{ or (b) } \mu X.A \in \mathbf{PF}).
\end{array}$$

We can easily check that **PF** and **NF** are closed under α-conversion of type expressions. Note that $\top \in$ **NF**.

Proposition 12.25. *Let A be a type expression, and X a type variable. If $X \in ETV^{\pm}(A)$, then there exist some A' and X' such that $A \simeq A'[X/X']$, $X' \in ETV^{\pm}(A')$ and $X' \notin ETV^{\mp}(A')$.*

Proof. By Proposition 5.23 and 5.5.1, we can assume that A is canonical. Note that $A \neq \top$ and $dp^{\pm}_{\to}(A,\, X) < \infty$ from $X \in ETV^{\pm}(A)$ by Proposition 4.13. The proof proceeds by straightforward induction on $dp^{\pm}_{\to}(A,\, X)$, and by cases of the form of A. $\square$

Proposition 12.26. *1. If $X \in ETV^{+}(A)$, $X \notin ETV^{-}(A)$ and $A[B/X] \in$ **PF** (**NF**), then $B \in$ **PF** (**NF**).*
*2. If $X \in ETV^{-}(A)$, $X \notin ETV^{+}(A)$ and $A[B/X] \in$ **PF** (**NF**), then $B \in$ **NF** (**PF**).*

Proof. By simultaneous induction on $h(A)$. Use Proposition 4.9.2 to show the part 2 in case that $A = C \to D$ for some C and D. Note that $X \in ETV^{\pm}(C \to D)$ implies that D is not a $\top$-variant; and if $A = \mu Y.C$ for some Y and D, then $Y \notin ETV^{-}(C)$ since $ETV^{+}(A) \neq ETV^{-}(A)$ by $X \in ETV^{+}(A)$ and $X \notin ETV^{-}(A)$ ($X \in ETV^{-}(A)$ and $X \notin ETV^{+}(A)$).

Proposition 12.27. **PF** $\subset$ **TF**.

Proof. We can show that $A \in$ **PF** implies $A \in$ **TF** for every A by straightforward induction on $h(A)$.

Lemma 12.28. *Suppose that*

(a) $A \in \mathbf{TF}^{V}$,
(b) $V \cap FTV(B) = \{\}$, *and*
(c) $X \notin ETV^{+}(A)$ *or* $B \in$ **TF**.

Then $A[B/X] \in \mathbf{TF}^{V}$.

Proof. If $X \notin ETV^{+}(A)$, then $A \in \mathbf{TF}^{V \cup \{X\}}$ by (a) and Proposition 12.16.2; and therefore, $A[B/X] \in \mathbf{TF}^{V}$ by Proposition 12.17.1. On the other hand, if $X \in ETV^{+}(A)$, then $B \in$ **TF** by (c); that is, $B \in \mathbf{TF}^{V}$ by (b) and Proposition 12.16.2. Hence, $A[B/X] \in \mathbf{TF}^{V}$ by Proposition 12.17.2. $\square$

Proposition 12.29. *Suppose that*

(a) $A \in$ **PF** (**NF**),

*(b) if $X \in ETV^+(A)$ then $B \in$ **PF** (**NF**), and*
*(c) if $X \in ETV^-(A)$ then $B \in$ **NF** (**PF**).*

*Then $A[B/X] \in$ **PF** (**NF**).*

Proof. By induction on $h(A)$, and by cases of the form of A. Use Proposition 4.9.1 if $A = C \to D$ for some C and D. The most interesting case is when $A = \mu Y.C$ for some Y and C. In this case, suppose that (a) through (c) hold. We can assume that $Y \notin FTV(B) \cup \{X\}$ without loss of generality; that is, $A[B/X] = \mu Y.C[B/X]$. By (a) and Definition 12.24, we have:

$$\mu Y.C \in \mathbf{TF}\ (\mathbf{TExp}) \tag{27}$$
$$C \in \mathbf{PF}\ (\mathbf{NF}),\ \text{and} \tag{28}$$
$$Y \notin ETV^-(C) \text{ or } C \in \mathbf{NF}\ (\mu Y.C \in \mathbf{PF}). \tag{29}$$

Besides, since $X \in ETV^{\pm}(C)$ implies $X \in ETV^{\pm}(A)$, from (b)and (c),

(b$'$) if $X \in ETV^+(C)$ then $B \in$ **PF** (**NF**), and
(c$'$) if $X \in ETV^-(C)$ then $B \in$ **NF** (**PF**).

By Definition 12.24, it suffices to show that:

$$\mu Y.C[B/X] \in \mathbf{TF}\ (\mathbf{TExp}) \tag{30}$$
$$C[B/X] \in \mathbf{PF}\ (\mathbf{NF}),\ \text{and} \tag{31}$$
$$Y \notin ETV^-(C[B/X]) \text{ or } C[B/X] \in \mathbf{NF}\ (\mu Y.C[B/X] \in \mathbf{PF}). \tag{32}$$

First, we get $C \in \mathbf{TF}^{\{Y\}}$ (C is proper in Y) from (27); and therefore, $C[B/X] \in \mathbf{TF}^{\{Y\}}$ ($C[B/X]$ is proper in Y) by (b$'$) and Lemma 12.28 (Proposition 4.7.2). Thus we get (30). Second, we get (31) from (28), (b$'$) and (c$'$) by the induction hypothesis. Finally, to show (32), suppose that $Y \in ETV^-(C[B/X])$. Then $Y \in ETV^-(C)$ by Proposition 4.10.2 since $Y \notin FTV(B)$; and therefore, $X \in ETV^{\pm}(C)$ implies $X \in ETV^{\mp}(A)$. Hence, from (b) and (c), we get:

(b$''$) if $X \in ETV^+(C)$ then $B \in$ **NF** (**PF**), and
(c$''$) if $X \in ETV^-(C)$ then $B \in$ **PF** (**NF**).

On the other hand, by $Y \in ETV^-(C)$ and (29), we get $C \in$ **NF** ($\mu Y.C \in$ **PF**; and therefore $C \in$ **PF** by Definition 12.24). Hence, $C[B/X] \in$ **NF** (**PF**) by the induction hypothesis. (Note that $C[B/X] \in$ **PF** implies $\mu Y.C[B/X] \in$ **PF** because $C[B/X] \in$ **NF** is already established as (31), and because we can get $\mu Y.C[B/X] \in$ **TF** from $\mu Y.C \in \mathbf{PF} \subset \mathbf{TF}$ and (b$''$) by Lemma 12.28.) We thus establish (32). □

Lemma 12.30. *If $A \in$ **PF** (**NF**), then A is positively (negatively) finite.*

Proof. It suffices to derive $C \in$ **TF** from the following assumptions:

(a) $A \in$ **PF** (**NF**),
(b) $A \simeq B[C/X]$,
(c) $X \in ETV^+(B)$ ($ETV^-(B)$), and
(d) $X \notin ETV^-(B)$ ($ETV^+(B)$).

By induction on the lexicographic ordering of $<dp^+_\bullet(B,\,X),\, r(A)>$ ($<dp^-_\bullet(B,\,X),\, r(A)>$), and by cases of the form of A. Suppose that (a) through (d) hold. We can assume that B is canonical because $B[C/X] \simeq B^c[C/X]$, $ETV^{\pm}(B) = ETV^{\pm}(B^c)$ and $dp^{\pm}_\bullet(B) = dp^{\pm}_\bullet(B^c)$ by Propositions 5.3, 5.23, 5.5.1 and 5.4. If $B = X$, then $X \notin ETV^-(B)$. In this case, we get $C \in$ **TF** by Propositions 12.18 and 12.13, because $C = B[C/X] \simeq A \in \mathbf{PF} \subset \mathbf{TF}$ by (a) and Proposition 12.27. Hence, we also assume that $B \neq X$.

Case: $A = P$ for some P, or $A = Y$ for some Y. This is impossible from the condition (c) and the assumption $B \neq X$.

Case: $A = \bullet D$ for some D. Since $B \neq X$, only possible case is when $B = \bullet D'$ for some D' such that $D \simeq D'[C/X]$. We get $D \in$ **PF** (**NF**) from (a); $X \in ETV^+(D')$ ($ETV^-(D')$) from (c); and $X \notin ETV^-(D')$ ($ETV^+(D')$) from (d). Since $dp^{\pm}_\bullet(B,\,X) = dp^{\pm}_\bullet(D',\,X) + 1$, we get $C \in$ **TF** by the induction hypothesis.

Case: $A = D \to E$ for some D and E. Since $B \not\equiv X$, only possible case is when $B = D' \to E'$ for some D' and E' such that $E \simeq E'[C/X]$, and $D \simeq D'[C/X]$ or $E \simeq \top$ by Propositions 5.15 and 5.14.2. If E is a $\top$-variant, i.e., $E \simeq \top$, then $A \notin$ **PF** and $E'[C/X]$ is also a $\top$-variant; and so is E' or $X \in ETV^+(E')$ by Proposition 4.9.2. However, this is impossible by (a), or (c) and (d). (Note that if E' is a $\top$-variant, then $X \notin ETV^-(D' \to E')$.) Therefore, E is not a $\top$-variant, i.e., $E \not\simeq \top$. We thus get:

- $D \in$ **NF** (**PF**), $D \simeq D'[C/X]$ and $X \notin ETV^+(D')$ $(ETV^-(D'))$.
- $E \in$ **PF** (**NF**), $E \simeq E'[C/X]$ and $X \notin ETV^-(E')$ $(ETV^+(E'))$.

On the other hand, we can get $dp^+_\bullet(B,\, X) < \infty$ $(dp^-_\bullet(B,\, X) < \infty)$ from (c) by Proposition 4.13.1; and since $dp^\pm_\bullet(B,\, X) = \min(dp^\mp_\bullet(D',\, X),\, dp^\pm_\bullet(E',\, X))$,

- $dp^+_\bullet(B,\, X) = dp^-_\bullet(D',\, X)$ $(dp^-_\bullet(B,\, X) = dp^+_\bullet(D',\, X))$ and $X \in ETV^-(D')$ $(ETV^+(D'))$, or
- $dp^+_\bullet(B,\, X) = dp^+_\bullet(E',\, X)$ $(dp^-_\bullet(B,\, X) = dp^-_\bullet(E',\, X))$ and $X \in ETV^+(E')$ $(ETV^-(E'))$.

Note also that $r(D),\, r(E) < r(A)$. Therefore, we get $C \in$ **TF** by the induction hypothesis in either case.

Case: $A = \mu Y.D$ for some Y and D. In this case, from (a),

$$\begin{aligned}
&\mu Y.D \in \textbf{TF}\ (\textbf{TExp}) \\
&D \in \textbf{PF}\ (\textbf{NF}),\ \text{and} && (33)\\
&Y \notin ETV^-(D) \text{ or } D \in \textbf{NF}\ (\mu Y.D \in \textbf{PF}). && (34)
\end{aligned}$$

It should be noted that (a), (33) and (34) imply that:

$$\text{if } Y \in ETV^-(D) \text{ then } \mu Y.D \in \textbf{NF}\ (\textbf{PF}). \tag{35}$$

We now get $D[\mu Y.D/Y] \in$ **PF** (**NF**) from (33), (a) and (35) by Proposition 12.29. Therefore, $C \in$ **TF** by the induction hypothesis, since $r(D[\mu Y.D/Y]) < r(A)$ and $D[\mu Y.D/Y] \simeq \mu Y.D = A \simeq B[C/X]$ from (b). □

Lemma 12.31. *If A is positively (negatively) finite, then $A \in$ **PF** (**NF**).*

Proof. By induction on $h(A)$, and by cases of the form of A. Suppose that A is positively (negatively) finite, i.e.:

$$\text{if } A \simeq B[C/X],\ X \in ETV^+(B)\ (ETV^-(B)), \text{ and } X \notin ETV^-(B)\ (ETV^+(B)), \text{ then } C \in \textbf{TF}. \tag{36}$$

Case: $A = P$ for some P, or $A = Y$ for some Y. Obviously, $A \in$ **PF** (**NF**) by Definition 12.24.

Case: $A = \bullet D$ for some D. Suppose that $D \simeq B'[C/X]$ and $X \in ETV^+(B')$ $(ETV^-(B'))$ hold for some X and B'. Since $A \simeq (\bullet B')[C/X]$ and $X \in ETV^+(\bullet B')$ $(ETV^-(\bullet B'))$, we get $C \in$ **TF** by (36). Therefore, $D \in$ **PF** (**NF**) by the induction hypothesis; and hence $A \in$ **PF** (**NF**) by Definition 12.24.

Case: $A = D \to E$ for some D and E. First, suppose that $D \simeq B'[C/X]$, $X \in ETV^-(B')$ $(ETV^+(B'))$ and $X \notin ETV^+(B')$ $(ETV^-(B'))$ hold for some X and B'. Since this means that $D \simeq B'[X'/X][C/X']$, $X' \in ETV^-(B'[X'/X])$ $(ETV^+(B'[X'/X]))$ and $X' \notin ETV^+(B'[X'/X])$ $(ETV^-(B'[X'/X]))$ for a fresh type variable X', we can assume that $X \notin FTV(E) \cup FTV(C)$ without loss of generality. We thus get $C \in$ **TF** by (36), since $A \simeq (B' \to E)[C/X]$, $X \in ETV^+(B' \to E)$ $(ETV^-(B' \to E))$ and $X \notin ETV^-(B' \to E)$ $(ETV^+(B' \to E))$; and therefore, $D \in$ **NF** (**PF**) by the induction hypothesis. We similarly get $E \in$ **PF** (**NF**). Hence $A \in$ **PF** (**NF**) by Definition 12.24.

Case: $A = \mu Y.D$ for some Y and D. If $A \simeq \top$, then obviously $A \in$ **NF**; and $A \simeq X[\top/X]$, which means that A is not positively finite. Therefore, we now assume that $A \not\simeq \top$, that is, A is not a $\top$-variant. It suffices to show the following:

$$\begin{aligned}
&\mu Y.D \in \textbf{TF}\ (\textbf{TExp}), && (37)\\
&D \in \textbf{PF}\ (\textbf{NF}),\ \text{and} && (38)\\
&Y \notin ETV^-(D) \text{ or } D \in \textbf{NF}\ (\mu Y.D \in \textbf{PF}). && (39)
\end{aligned}$$

First, if A is positively finite, then we get $\mu Y.D \in$ **TF** from (36) since $A \simeq X[\mu Y.D/X]$. Second, to show (38), suppose that the following hold for some X and B':

$$D \simeq B'[C/X] \tag{40}$$

$$X \in ETV^+(B')\ (ETV^-(B')),\ \text{and} \tag{41}$$

$$X \notin ETV^-(B')\ (ETV^+(B')). \tag{42}$$

We can assume that $X \notin FTV(A) \cup \{Y\}$ without loss of generality. Therefore, we get $A \simeq \mu Y.B'[C/X] \simeq B'[C/X][\mu Y.B'[C/X]/Y] \simeq B'[C/X][A/Y] = B'[C[A/Y]/X, A/Y] = B'[A/Y][C[A/Y]/X]$ from (40) by Proposition 5.8.1, Proposition 5.3 and ($\simeq$-fix); and $X \in ETV^+(B'[A/Y])\ (ETV^-(B'[A/Y]))$ from (41) by Proposition 4.10.1 since A is not a $\top$-variant. Note also that we can get $X \notin ETV^-(B'[A/Y])\ (ETV^+(B'[A/Y]))$ from (42) by Proposition 4.10.2 since $X \notin FTV(A)$. Therefore, we get $C[A/Y] \in$ **TF** by (36); and hence $C \in$ **TF** by Proposition 12.17.3. We now get $D \in$ **PF** (**NF**) by the induction hypothesis. Finally, to establish (39), suppose that $Y \in ETV^-(D)$, i.e., by Proposition 12.25, there exist some D' and Y' such that:

$$D \simeq D'[Y/Y'], \tag{43}$$

$$Y' \in ETV^-(D'),\ \text{and} \tag{44}$$

$$Y' \notin ETV^+(D'). \tag{45}$$

We can assume that $Y' \notin FTV(D) \cup \{Y\}$ without loss of generality. If D is negatively (positively) finite, then we get $D \in$ **NF** (**PF**) by the induction hypothesis. (Note that if A is negatively finite, then $D \in$ **PF** implies $\mu Y.D \in$ **PF**, because $D \in$ **NF** is already established as (38), and because we can get $\mu Y.D \in$ **TF** by (36) from (44), (45) and $A = \mu Y.D \simeq \mu Y.D'[Y/Y'] \simeq \mu Y.\mu Y'.D' \simeq D'[A/Y', A/Y] \simeq D'[A/Y][A/Y']$.) That is, to establish (39), it suffices to show that D is negatively (positively) finite. So suppose that there exist some B'' and X such that:

$$D \simeq B''[C/X], \tag{46}$$

$$X \in ETV^-(B'')\ (ETV^+(B'')),\ \text{and} \tag{47}$$

$$X \notin ETV^+(B'')\ (ETV^-(B'')). \tag{48}$$

We can assume that $X \notin FTV(A) \cup FTV(D') \cup \{Y, Y'\}$.

$$\begin{aligned}
A &\simeq \mu Y.D'[Y/Y'] && \text{(by (43) and Proposition 5.8.1)}\\
&\simeq \mu Y.D'[D'[Y/Y']/Y'] && \text{(by Proposition 5.8.3)}\\
&\simeq \mu Y.D'[D/Y'] && \text{(by (43) and Proposition 5.3)}\\
&\simeq \mu Y.D'[B''[C/X]/Y'] && \text{(by (46) and Proposition 5.3)}\\
&\simeq D'[B''[C/X]/Y'][\mu Y.D'[B''[C/X]/Y']/Y] && \text{(by } (\simeq\text{-fix}))\\
&\simeq D'[B''[C/X]/Y'][A/Y] && \text{(by Proposition 5.3)}\\
&= D'[B''[C/X][A/Y]/Y', A/Y] && \text{(since } Y \neq Y')\\
&= D'[B''[A/Y][C[A/Y]/X]/Y', A/Y] && \text{(since } X \notin FTV(A) \cup \{Y\})\\
&= D'[B''[A/Y]/Y', A/Y][C[A/Y]/X] && \text{(since } X \notin FTV(A) \cup FTV(D'))\\
&= D'[B''/Y'][A/Y][C[A/Y]/X] && \text{(since } Y \neq Y')
\end{aligned}$$

We get $X \in ETV^+(D'[B''/Y'])\ (ETV^-(D'[B''/Y']))$ from (44) and (47) by Proposition 4.11.2; and hence,

$$X \in ETV^+(D'[B''/Y'][A/Y]) \quad (X \in ETV^-(D'[B''/Y'][A/Y]))$$

by Proposition 4.10.1 since $Y \neq X$ and A is not a $\top$-variant. Similarly, we get $X \notin ETV^-(D'[B''/Y'])$ $(ETV^+(D'[B''/Y']))$ from (45), (48) and $X \notin FTV(D')$ by Proposition 4.11.3; and

$$X \notin ETV^-(D'[B''/Y'][A/Y]) \quad (X \notin ETV^+(D'[B''/Y'][A/Y]))$$

by Proposition 4.10.2 since $X \notin FTV(A)$. Hence, $C[A/Y] \in$ **TF** by (36); and $C \in$ **TF** by Proposition 12.17.3. We thus get the fact that D is negatively (positively) finite, and (39) is now established. $\square$

Proposition 12.32. *A is positively (negatively) finite if and only if $A \in$* **PF** *(***NF***).*

Proof. Straightforward from Lemmas 12.30 and 12.31. □

Lemma 12.33. *Let A and B be type expressions such that $A \preceq B$.*

1. *If B is positively finite, then so is A.*
2. *If A is negatively finite, then so is B.*

Proof. It suffices to show that for every X, A, B, C and D, D is tail finite provided the following:

(a) $dp^{+}_{\rightarrow}(C, X) < \infty$, $dp^{-}_{\rightarrow}(C, X) = \infty$, $C[D/X] \preceq B$ and B is positively finite, or
(b) $dp^{-}_{\rightarrow}(C, X) < \infty$, $dp^{+}_{\rightarrow}(C, X) = \infty$, $A \preceq C[D/X]$ and A is negatively finite.

The proof proceeds by induction on $dp^{\pm}_{\rightarrow}(C, X)$. Suppose that (a) or (b) holds. We can assume that C is canonical because $ETV^{\pm}(C) = ETV^{\pm}(C^c)$, $C[D/X] \simeq C^c[D/X]$ and $dp^{\pm}_{\rightarrow}(C, X) = dp^{\pm}_{\rightarrow}(C^c, X)$ by Propositions 5.23, 5.5.1, 5.3 and 5.4.

Case: $C = \top$ or $C = \bullet^n P$ for some n and P. This is impossible because $X \in FTV(C)$ from (a) or (b) by Proposition 4.13.

Case: $C = \bullet^n Y$ for some Y. Note that $dp^{-}_{\rightarrow}(C, X) = \infty$ in this case. If (a) holds, then $Y = X$ since $dp^{+}_{\rightarrow}(C, X) < \infty$. Therefore, D is tail finite by Proposition 12.13, because $tl_{\{\}}(D) = tl_{\{\}}(\bullet^n D) = tl_{\{\}}(C[D/X]) \leq tl_{\{\}}(B)$ by Propositions 12.12 and 12.13.

Case: $C = \bullet^n(E \rightarrow F)$ for some E and F. In this case, $C[D/X] = \bullet^n(E \rightarrow F)[D/X] = \bullet^n(E[D/X] \rightarrow F[D/X])$. Note that $F \not\simeq \top$ since $dp^{+}_{\rightarrow}(C, X) < \infty$ or $dp^{-}_{\rightarrow}(C, X) < \infty$ holds. If (a) holds, then $B \not\simeq \top$ since B is positively finite; and we get:

- $dp^{-}_{\rightarrow}(E, X) < dp^{+}_{\rightarrow}(C, X)$ or $dp^{+}_{\rightarrow}(F, X) < dp^{+}_{\rightarrow}(C, X)$,
- $dp^{+}_{\rightarrow}(E, X) = dp^{-}_{\rightarrow}(F, X) = \infty$, and
- $\bullet^n(E[D/X] \rightarrow F[D/X]) \preceq B$,

because $dp^{\pm}_{\rightarrow}(C, X) = dp^{\pm}_{\rightarrow}(\bullet^n(E \rightarrow F)) = \min(dp^{\mp}_{\rightarrow}(E, X), dp^{\pm}_{\rightarrow}(F, X)) + 1$ by Definition 4.12. Since $B \not\simeq \top$, by Proposition 6.9, there exist some m, k, l, G and H such that:

(a1) $B \simeq \bullet^m(G \rightarrow H)$,
(a2) $n+k \leq m$,
(a3) $\bullet^k G \preceq \bullet^l E[D/X]$ and $\bullet^l F[D/X] \preceq \bullet^k H$.

Since B is positively finite, G and H are negatively and positively finite, respectively; and so are $\bullet^k G$ and $\bullet^k H$. Therefore, D is tail finite from (a3) by the induction hypothesis because $dp^{+}_{\rightarrow}(\bullet^l E, X) = dp^{+}_{\rightarrow}(E, X) = \infty$, $dp^{-}_{\rightarrow}(\bullet^l F, X) = dp^{-}_{\rightarrow}(F, X) = \infty$, and $dp^{-}_{\rightarrow}(\bullet^l E, X) = dp^{-}_{\rightarrow}(E, X) < dp^{+}_{\rightarrow}(C, X)$ or $dp^{+}_{\rightarrow}(\bullet^l F, X) = dp^{+}_{\rightarrow}(F, X) < dp^{+}_{\rightarrow}(C, X)$. On the other hand, if (b) holds, then we similarly get:

- $dp^{+}_{\rightarrow}(E, X) < dp^{-}_{\rightarrow}(C, X)$ or $dp^{-}_{\rightarrow}(F, X) < dp^{-}_{\rightarrow}(C, X)$,
- $dp^{-}_{\rightarrow}(E, X) = dp^{+}_{\rightarrow}(F, X) = \infty$, and
- $A \preceq \bullet^n(E[D/X] \rightarrow F[D/X])$.

We get $X \notin ETV^{+}(F)$ from $dp^{+}_{\rightarrow}(F, X) = \infty$; and hence, $F[D/X] \not\simeq \top$ from $F \not\simeq \top$ by Proposition 4.9. Therefore, by Proposition 6.9, there exist some m, k, l, G and H such that:

(b1) $A \simeq \bullet^m(G \rightarrow H)$,
(b2) $m+k \leq n$,
(b3) $\bullet^l H \preceq \bullet^k F[D/X]$, and
(b4) $\bullet^k E[D/X] \preceq \bullet^l G$.

Since A is negatively finite, G and H are positively and negatively finite, respectively; and so are $\bullet^l G$ and $\bullet^l H$. Therefore, D is tail finite from (b3) and (b4) by the induction hypothesis because $dp^{-}_{\rightarrow}(\bullet^k E, X) = dp^{-}_{\rightarrow}(E, X) = \infty$, $dp^{+}_{\rightarrow}(\bullet^k F, X) = dp^{+}_{\rightarrow}(F, X) = \infty$, and $dp^{+}_{\rightarrow}(\bullet^k E, X) = dp^{+}_{\rightarrow}(E, X) < dp^{-}_{\rightarrow}(C, X)$ or $dp^{-}_{\rightarrow}(\bullet^k F, X) = dp^{-}_{\rightarrow}(F, X) < dp^{-}_{\rightarrow}(C, X)$.

□

Lemma 12.34. *Let n be a non-negative integer, and $x_1, x_2, \ldots, x_n$ distinct type variables. If $\Gamma \vdash \lambda x_1. \lambda x_2. \ldots \lambda x_n. M : A$ is derivable, then so is $\Gamma \cup \{x_1 : B_1, x_2 : B_2, \ldots, x_n : B_n\} \vdash M : C$ for some B_1, B_2, …, B_n and C such that $B_1 \rightarrow B_2 \rightarrow \ldots \rightarrow B_n \rightarrow C \preceq A$.*

Proof. By induction on n. It is trivial if $n = 0$. If $n > 0$, then there are only two possible cases.

Case 1. The derivation ends as follows:

$$\cfrac{\cfrac{}{\Gamma \vdash \lambda x_1.\,\lambda x_2.\,\ldots\lambda x_n.\,M : \top}\,(\top)}{\vdots \text{ 0 or more } (\preceq)\text{'s} \atop \Gamma \vdash \lambda x_1.\,\lambda x_2.\,\ldots\lambda x_n.\,M : A}$$

In this case, $\top \preceq A$; and therefore, $A \simeq \top$ by Proposition 6.5. We can derive $\Gamma \cup \{x_1 : B_1,\, x_2 : B_2,\, \ldots,\, x_n : B_n\} \vdash M : \top$ by the $(\top)$-rule, and $B_1 \to B_2 \to \ldots \to B_n \to \top \preceq \top$ by $(\preceq\text{-}\top)$ for any $B_1, B_2, \ldots, B_n$.

Case 2. There exist some B_1 and D such that $B_1 \to D \preceq A$ and the derivation ends with:

$$\cfrac{\cfrac{\vdots \atop \Gamma \cup \{x_1 : B_1\} \vdash \lambda x_2.\,\ldots\lambda x_n.\,M : D}{\Gamma \vdash \lambda x_1.\,\lambda x_2.\,\ldots\lambda x_n.\,M : B_1 \to D}\,(\to\mathrm{I})}{\vdots \text{ 0 or more } (\preceq)\text{'s} \atop \Gamma \vdash \lambda x_1.\,\lambda x_2.\,\ldots\lambda x_n.\,M : A}$$

In this case, by the induction hypothesis, $\Gamma \cup \{x_1 : B_1,\, x_2 : B_2,\, \ldots,\, x_n : B_n\} \vdash M : C$ is derivable for some $B_2, \ldots, B_n$ and C such that $B_2 \to \ldots \to B_n \to C \preceq D$. Therefore, $B_1 \to B_2 \to \ldots \to B_n \to C \preceq B_1 \to D \preceq A$. ◻

Lemma 12.35. *Suppose that $A \not\simeq \top$. If $\Gamma \vdash xN_1N_2\ldots N_n : A$ is derivable, then $\Gamma(x) \simeq \bullet^{m_1}(B_1 \to \bullet^{m_2}(B_2 \to \ldots \to \bullet^{m_n}(B_n \to C)\ldots))$ 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 ($0 \le i \le n$), there exists some m'_i such that $\Gamma \vdash N_i\, \bullet^{m'_i}B_i$ is derivable.*

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

$$\cfrac{\cfrac{}{\Gamma \vdash x : \Gamma(x)}\,(\mathrm{var})}{\vdots \text{ 0 or more } (\preceq)\text{'s} \atop \Gamma \vdash x : A}$$

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:

$$\cfrac{\cfrac{\vdots \atop \Gamma \vdash xN_1N_2\ldots N_{n-1} : \bullet^{m'}(D \to E) \qquad \vdots \atop \Gamma \vdash N_n : \bullet^{m'}D}{\Gamma \vdash xN_1N_2\ldots N_n : \bullet^{m'}E}\,(\to\mathrm{E})}{\vdots \text{ 0 or more } (\preceq)\text{'s} \atop \Gamma \vdash xN_1N_2\ldots N_n : A}$$

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

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

Therefore, since $E \not\simeq \top$, by Proposition 6.9, there exist some m'', k, l, B_n and C such that:

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

We get $C' \preceq \bullet^{m_n}(B_n \to C)$ by Propositions 5.13 and 5.15, where $m_n = m'' - m_1 - m_2 - \ldots - m_{n-1}$; and therefore, $\Gamma(x) \simeq \bullet^{m_1}(B_1 \to \bullet^{m_2}(B_2 \to \ldots \to \bullet^{m_{n-1}}(B_{n-1} \to \bullet^{m_n}(B_n \to C)\ldots)))$. On the other hand, $\bullet^{m_1+m_2+\ldots+m_{n-1}+m_n} C = \bullet^{m''} C \preceq \bullet^{m'-k} C \preceq \bullet^{m'-k+l} C \preceq \bullet^{m'} E \preceq A$ and $\bullet^{m'} D \preceq \bullet^{m'-k+l} B_n$. We get the derivation of $\Gamma \vdash N_n : \bullet^{m'-k+l} B_n$ from the one of $\Gamma \vdash N_n : \bullet^{m'} D$ by $(\preceq)$. □

Theorem 12.36. *Let $\Gamma \vdash M : A$ be a derivable judgment. If A is positively finite and $\Gamma(x)$ is negatively finite for every $x \in Dom(\Gamma)$, then M is maximal.*

Proof. 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 Theorem 12.21, 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 $x_1, x_2, \ldots, x_m, y, N_1, N_2, \ldots, N_l$. By Theorem 8.5, $\Gamma \vdash \lambda x_1.\, \lambda x_2.\, \ldots \lambda x_m.\, y N_1\, N_2\, \ldots\, N_l : A$ is also derivable; and therefore, by Lemma 12.34, $\Gamma \cup \{x_1 : B_1, x_2 : B_2, \ldots, x_m : B_m\} \vdash y N_1\, N_2\, \ldots\, N_l : C$ is derivable for some $B_1, B_2, \ldots, B_m$ and C such that:

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

Note that 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 12.33. Let $\Gamma' = \Gamma \cup \{x_1 : B_1, x_2 : B_2, \ldots, x_m : B_m\}$. Since $C \not\simeq \top$, by Lemma 12.35, $\Gamma'(y) \simeq \bullet^{k_1}(D_1 \to \bullet^{k_2}(D_2 \to \ldots \to \bullet^{k_l}(D_l \to E)\ldots))$ for some $k_1, k_2, \ldots, k_l, D_1, D_2, \ldots, D_l$ and E such that:

- $\bullet^{k_1+k_2+\ldots+k_l} E \preceq C$, and
- for every i $(0 \le i \le l)$, $\Gamma' \vdash N_i : \bullet^{k'_i} D_i$ 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 $(0 \le i \le l)$ because $\Gamma'(z)$ is negatively finite for every $z \in Dom(\Gamma)$. Therefore, by the induction hypothesis, for every i $(0 \le i \le l)$, every node of the Böhm-tree of N_i at the level less than n is head normalizable; that is, so is one of M at the level less than or equal to n. □

13 $\lambda\bullet\mu$ as a logic

We can regard the typing system $\lambda\bullet\mu$ as a logical system by ignoring left hand sides of : from typing judgments. In this section, we discuss the some properties of $\lambda\bullet\mu$ as such a logical system.

13.1 Redundancy of the $(\bullet)$-rule

In this subsection, we show that the rule $(\bullet)$ is redundant when we regard $\lambda\bullet\mu$ as a logic.

Definition 13.1. Let A be a type expression. We define $\overline{A}$ by induction on $r(A)$ as follows:

$$\overline{P} = P, \qquad \overline{X} = X,$$
$$\overline{\bullet A} = A, \qquad \overline{A \to B} = \begin{cases} A \to \top & (B \text{ is a } \top\text{-variant}) \\ A \to \overline{A} \to \overline{B} & (\text{otherwise}), \end{cases}$$
$$\overline{\mu X.A} = \overline{A}[\mu X.A/X].$$

Proposition 13.2. *If A is proper in X, then $\overline{A[B/X]} = \overline{A}[B/X]$.*

Proof. By straightforward induction on $h(A)$. □

Proposition 13.3. *If $A \simeq B$, then $\overline{A} \simeq \overline{B}$.*

Proof. By straightforward induction on the derivation of $A \simeq B$. Use Proposition 13.2 for the case ($\simeq$-fix) and ($\simeq$-uniq). □

Proposition 13.4. *Let A, B, $C_1, C_2, \ldots, C_n$ $D_1, D_2, \ldots, D_n$ be type expressions. Let $X_1, X_2, \ldots, X_n$ $Y_1, Y_2, \ldots, Y_n$ be distinct type variables. If $\{X_1 \preceq Y_1, X_2 \preceq Y_2, \ldots, X_n \preceq Y_n\} \vdash A \preceq B$ is derivable, and $C_i \preceq D_i$ for every i $(0 \le i \le n)$, then $\vdash M : (\overline{A \to B})\sigma$ is derivable without using the $(\bullet)$-rule for some λ-term M, where $\sigma = [C_1/X_1, D_1/Y_1, C_2/X_2, D_2/Y_2, \ldots, C_n/X_n, D_n/Y_n]$.*

Proof. By induction on the derivation of $\{X_1 \preceq Y_1, X_2 \preceq Y_2, \ldots, X_n \preceq Y_n\} \vdash A \preceq B$, and by cases of the last rule applied in the derivation. First note that $A\sigma \preceq B\sigma$ by Proposition 6.4.

Cases: ($\preceq$*-assump*), ($\preceq$*-reflex*) *and* ($\preceq$*-*$\bullet$). In these cases, we can take $\lambda x.\, \lambda y.\, y$ for M, because $\overline{A} \preceq \overline{B}$. Use Proposition 13.3 in the case ($\preceq$-reflex).

Case: ($\preceq$-$\top$). In this case, $B = \top$; therefore, $\overline{A \to B}\sigma = (A \to \top)\sigma = A\sigma \to \top$. We can derive $\vdash \lambda x.x : A\sigma \to \top$ as follows:

$$\dfrac{\dfrac{}{x : A\sigma \;\vdash\; x : \top}\;(\top)}{\vdash\; \lambda x.\, x : A\sigma \to \top}\;(\to\mathrm{I})$$

Case: ($\preceq$*-trans*). The derivation ends with:

$$\dfrac{\begin{matrix}\vdots \\ \gamma_1 \;\vdash\; A \preceq E\end{matrix} \qquad \begin{matrix}\vdots \\ \gamma_2 \;\vdash\; E \preceq B\end{matrix}}{\gamma_1 \cup \gamma_2 \;\vdash\; A \preceq B}\;(\preceq\text{-trans})$$

for some E, γ_1 and γ_2 such that $\gamma = \gamma_1 \cup \gamma_2$. We can assume that $\gamma_1 = \gamma_2 = \gamma$ by Proposition 6.3.3. By the induction hypothesis, we have derivations of $\vdash K : A\sigma \to \overline{A}\sigma \to \overline{E}\sigma$ and $\vdash L : E\sigma \to \overline{E}\sigma \to \overline{B}\sigma$ for some K and L. Therefore, we can derive $\vdash \lambda x.\, \lambda y.\, Lx(Kx\,y) : A\sigma \to \overline{A}\sigma \to \overline{B}\sigma$. Note that $A\sigma \preceq E\sigma$.

Case: ($\preceq$-$\to$). The derivation ends with:

$$\dfrac{\begin{matrix}\vdots \\ \gamma_1 \;\vdash\; G \preceq E\end{matrix} \qquad \begin{matrix}\vdots \\ \gamma_2 \;\vdash\; F \preceq H\end{matrix}}{\gamma_1 \cup \gamma_2 \;\vdash\; E \to F \preceq G \to H}\;(\preceq\text{-}\to)$$

for some E, F, G, H, γ_1 and γ_2 such that $A = E \to F$, $B = G \to H$, and $\gamma = \gamma_1 \cup \gamma_2$. That is, $A\sigma \to \overline{A}\sigma \to \overline{B}\sigma = (E\sigma \to F\sigma) \to (E\sigma \to \overline{E}\sigma \to \overline{F}\sigma) \to G\sigma \to \overline{G}\sigma \to \overline{H}\sigma$. We assume that $\gamma_1 = \gamma_2 = \gamma$ by Proposition 6.3.3, again. By the induction hypothesis, we have derivations of $\vdash K : G\sigma \to \overline{G}\sigma \to \overline{E}\sigma$ and $\vdash L : F\sigma \to \overline{F}\sigma \to \overline{H}\sigma$ for some K and L. Therefore, we can derive $\vdash \lambda f.\lambda g.\lambda x.\lambda y.L(f x)\,(gx(Kx\,y)) : (E\sigma \to F\sigma) \to (E\sigma \to \overline{E}\sigma \to \overline{F}\sigma) \to G\sigma \to \overline{G}\sigma \to \overline{H}\sigma$. Note that $G\sigma \preceq E\sigma$.

Case: ($\preceq$*-approx*). In this case, $B = \bullet A$. Hence, $A\sigma \to \overline{A}\sigma \to \overline{B}\sigma = A\sigma \to \overline{A}\sigma \to A\sigma$. Therefore, we can take $\lambda x.\, \lambda y.\, x$ for M.

Case: ($\preceq$-$\to\bullet$). That is, $A = E \to F$ and $B = \bullet E \to \bullet F$ for some E and F. We can take $\lambda f.\, \lambda g.\, \lambda x.\, \lambda y.\, f y$ for M, because $A\sigma \to \overline{A}\sigma \to \overline{B}\sigma = (E\sigma \to F\sigma) \to (E\sigma \to \overline{E}\sigma \to \overline{F}\sigma) \to \bullet E\sigma \to E\sigma \to F\sigma$.

Case: ($\preceq$-$\bullet\to$). $A = \bullet E \to \bullet F$ and $B = \bullet(E \to F)$ for some E and F. We can take $\lambda f.\, \lambda g.\, \lambda x.\, gxx$ for M, because $A\sigma \to \overline{A}\sigma \to \overline{B}\sigma = (\bullet E\sigma \to \bullet F\sigma) \to (\bullet E\sigma \to E\sigma \to F\sigma) \to E\sigma \to F\sigma$.

Case: ($\preceq$-μ). The derivation ends with:

$$\dfrac{\begin{matrix}\vdots \\ \gamma \cup \{X \preceq Y\} \;\vdash\; A' \preceq B'\end{matrix}}{\gamma \;\vdash\; \mu X.A' \preceq \mu Y.B'}\;(\preceq\text{-}\mu)$$

for some X, Y, A' and B' such that $X \notin FTV(B')$, $Y \notin FTV(A')$, $A = \mu X.A'$ and $B = \mu Y.B'$. We get $(\mu X.A')\sigma \preceq (\mu Y.B')\sigma$ by Proposition 6.4. Therefore, by the induction hypothesis, we have a derivation of $\vdash M : A'\sigma' \to \overline{A'}\sigma' \to \overline{B'}\sigma'$ for some M, where $\sigma' = [(\mu X.A)\sigma/X,\, (\mu Y.B)\sigma/Y,\, C_1/X_1,\, D_1/Y_1,\, C_2/X_2,\, D_2/Y_2,\, \ldots,\, C_n/X_n,\, D_n/Y_n]$. Since obviously $FTV(A') = FTV(\overline{A'})$ and $FTV(B') = FTV(\overline{B'})$, we get $X \notin FTV(\overline{B'})$ and $Y \notin FTV(\overline{A'})$. Hence,

$$\begin{aligned}
A'\sigma' &= A'[\mu X.A'/X]\sigma \simeq (\mu X.A')\sigma = A\sigma \\
\overline{A'}\sigma' &= \overline{A'}[\mu X.A'/X]\sigma \simeq \overline{\mu X.A'}\sigma = \overline{A}\sigma \\
\overline{B'}\sigma' &= \overline{B'}[\mu Y.B'/Y]\sigma \simeq \overline{\mu Y.B'}\sigma = \overline{B}\sigma
\end{aligned}$$

That is, $A'\sigma' \to \overline{A'}\sigma' \to \overline{B'}\sigma' \simeq A\sigma \to \overline{A}\sigma \to \overline{B}\sigma$. Therefore, $\vdash M : A\sigma \to \overline{A}\sigma \to \overline{B}\sigma$ is also derivable. $\square$

Proposition 13.5. *Let $x_1, y_1, x_2, y_2, \ldots, x_n, y_n$ be distinct individual variables. If $\{x_1 : A_1, x_2 : A_2 \ldots, x_n : A_n\} \vdash M : B$ is derivable, then we can derive*

1. $\{x_1 : A_1, x_2 : A_2, \ldots, x_n : A_n\} \vdash K : B$*, and*
2. $\{x_1 : A_1, y_1 : \overline{A_1}, x_2 : A_2, y_2 : \overline{A_2}, \ldots, x_n : A_n, y_n : \overline{A_n}\} \vdash L : \overline{B}$

without using the (•)-rule for some λ-terms K and L

Proof. Let Π be the derivation of $\{x_1 : A_1, x_2 : A_2 \ldots, x_n : A_n\} \vdash M : B$. Let $l(\Pi)$ be the number of occurrences of the (•)-rule in the derivation Π, and $c(\Pi)$ the number of occurrences of any typing rule in Π. We prove the conjecture by induction on the lexicographic ordering of $<l(\Pi), c(\Pi)>$ and by cases of the last rule applied in Π. Let $\Gamma = \{x_1 : A_1, x_2 : A_2, \ldots, x_n : A_n\}$ and $\Gamma' = \{x_1 : A_1, y_1 : \overline{A_1}, x_2 : A_2, y_2 : \overline{A_2}, \ldots, x_n : A_n, y_n : \overline{A_n}\}$. Note that if B is a $\top$-variant, then so is $\overline{B}$; and hence, $\Gamma \vdash K : B$ and $\Gamma' \vdash L : \overline{B}$ are derivable for any K and L. We therefore assume that B is not a $\top$-variant, i.e., $B \not\simeq \top$ in the sequel.

Case: (*var*). Obvious because $B = A_i$ for some i ($0 \le i \le n$).

Case: (*const*). Obvious because $B = \overline{B}$.

Case: ($\top$). Impossible from the assumption that $B \not\simeq \top$.

Case: (•). The derivation ends with:

$$\dfrac{\begin{array}{c}\vdots\\ \{x_1 : \bullet A_1, x_2 : \bullet A_2, \ldots, x_n : \bullet A_n\} \vdash M : \bullet B\end{array}}{\{x_1 : A_1, x_2 : A_2, \ldots, x_n : A_n\} \vdash M : B}\ (\bullet)$$

By the induction hypothesis, $\{x_1 : \bullet A_1, y_1 : A_1, x_2 : \bullet A_2, y_2 : A_2, \ldots, x_n : \bullet A_n, y_n : A_n\} \vdash L : B$ is derivable without using the (•)-rule for some L. Therefore, there is some derivation Π' of $\Gamma \vdash L[x_1/y_1, x_2/y_2, \ldots, x_n/y_n] : B$ which includes no (•)-rule. On the other hand, since $c(\Pi') < c(\Pi)$, by the induction hypothesis again, $\Gamma' \vdash L' : \overline{B}$ is derivable without using the (•)-rule for some L'.

Case: ($\preceq$). For some B', the derivation ends with:

$$\dfrac{\begin{array}{c}\vdots\\ \Gamma \vdash M : B'\end{array} \qquad \begin{array}{c}\vdots\\ \vdash B' \preceq B\end{array}}{\Gamma \vdash M : B}\ (\bullet)$$

By the induction hypothesis, $\Gamma \vdash K : B'$ and $\Gamma' \vdash L : \overline{B'}$ are derivable without using the (•)-rule for some K and L; and so is $\Gamma \vdash K : B$ because $B' \preceq B$. On the other hand, since $B \not\simeq \top$, by Proposition 13.4, $\vdash N : B' \to \overline{B'} \to \overline{B}$ is derivable without using the (•)-rule for some N. Hence, so is $\Gamma' \vdash N K L : \overline{B}$.

Case: ($\to$*I*). For some z, N, B_1 and B_2 such that $z \neq Dom(\Gamma)$, $M = \lambda z.\, N$ and $B = B_1 \to B_2$, the derivation ends with:

$$\dfrac{\begin{array}{c}\vdots\\ \Gamma \cup \{z : B_1\} \vdash N : B_2\end{array}}{\Gamma \vdash \lambda z.\, N : B_1 \to B_2}\ (\to\mathrm{I})$$

We can assume that $z \notin Dom(\Gamma')$. Let z' be a fresh individual variable. By the induction hypothesis, $\Gamma \cup \{z : B_1\} \vdash K : B_2$ and $\Gamma' \cup \{z : B_1, z' : \overline{B_1}\} \vdash L : \overline{B_2}$ are derivable without using the (•)-rule for some K and L; and so are $\Gamma \vdash \lambda z.\, K : B_1 \to B_2$ and $\Gamma' \vdash \lambda z.\, \lambda z'.\, L : B_1 \to \overline{B_1} \to \overline{B_2}$. Note that if $B_2 \not\simeq \top$, then $\overline{B} = B_1 \to \overline{B_1} \to \overline{B_2}$; and otherwise, $\overline{B} = B_1 \to \top$. Therefore, $\Gamma' \vdash \lambda z.\, \lambda z'.\, L : \overline{B}$ is derivable without using the (•)-rule in either case.

Case: $(\rightarrow E)$. For some n, M_1, M_2, B' and C such that $M = M_1 M_2$ and $B = \bullet^n B'$, the derivation ends with:

$$\dfrac{\begin{matrix}\vdots & & \vdots \\ \Gamma \vdash M_1 : \bullet^n(C \rightarrow B') & & \Gamma \vdash M_2 : \bullet^n C\end{matrix}}{\Gamma \vdash M_1 M_2 : \bullet^n B'} \ (\rightarrow\mathrm{E})$$

By the induction hypothesis, $\Gamma \vdash K_1 : \bullet^n(C \rightarrow B')$, $\Gamma \vdash K_2 : \bullet^n C$, $\Gamma' \vdash L_1 : \overline{\bullet^n(C \rightarrow B')}$ and $\Gamma' \vdash L_2 : \overline{\bullet^n C}$ are derivable without using the $(\bullet)$-rule for some K_1, K_2, L_1 and L_2. Therefore, so is $\Gamma \vdash K_1 K_2 : \bullet^n B'$. If $n = 0$, i.e., $B = B' \not\simeq \top$, then we can derive $\Gamma' \vdash L_1 K_2 L_2 : \overline{B}$ without using the $(\bullet)$-rule because $\overline{\bullet^n(C \rightarrow B')} = C \rightarrow \overline{C} \rightarrow \overline{B}$ and $\overline{\bullet^n C} = \overline{C}$. On the other hand, if $n > 0$, we can derive $\Gamma' \vdash L_1 L_2 : \overline{\bullet^n B'}$ without using the $(\bullet)$-rule because $\overline{\bullet^n B'} = \bullet^{n-1} B'$, $\overline{\bullet^n(C \rightarrow B')} = \bullet^{n-1}(C \rightarrow B')$ and $\overline{\bullet^n C} = \bullet^{n-1} C$. $\square$

Remark 13.6. Proposition 13.5 says that the existence of the $(\bullet)$-rule does not affects type inhabitance, i.e., the $(\bullet)$-rule is redundant if we regard $\lambda\bullet\mu$ as a logic ignoring left hand sides of :. However, the $(\bullet)$-rule is not redundant as a typing system. For example, let $M = (\lambda f.\, y(f(\lambda z.\, x))\,(g f))\,(\lambda h.\, h(\lambda u.\, u))$. While we can derive $\{x : \bullet X,\, y : \bullet(X \rightarrow Y \rightarrow Z),\, g : \bullet((((Y \rightarrow Y) \rightarrow \bullet X) \rightarrow \bullet X) \rightarrow Y)\} \vdash M : \bullet Z$ in $\lambda\bullet\mu$, we can not derive $\{x : X,\, y : X \rightarrow Y \rightarrow Z,\, g : (((Y \rightarrow Y) \rightarrow \bullet X) \rightarrow \bullet X) \rightarrow Y\} \vdash M : Z$ without using the $(\bullet)$-rule.

Remark 13.7. Let M be a β-normal form. If $\Gamma \vdash M : A$ is derivable, then we can also derive it without using the $(\bullet)$-rule.

Remark 13.8. The proof of Proposition 13.5 also suggests that if we had intersection types, the $(\bullet)$-rule would be redundant even as a typing system. If we change the definition of $\overline{A \rightarrow B}$ as $\overline{A \rightarrow B} = (A \wedge \overline{A}) \rightarrow \overline{B}$ for non-$\top$-variant B, then we get the following:

1. If $A \simeq B$, then $\overline{A} \simeq \overline{B}$.
2. If $A \preceq B$, then $\overline{A} \preceq \overline{B}$.
3. If $\Gamma \vdash M : A$ is derivable in $\lambda\bullet\mu$ + "intersection types", then we can also derive $\Gamma \wedge \overline{\Gamma} \vdash M : \overline{B}$ in it without using the $(\bullet)$-rule, where $Dom(\Gamma \wedge \overline{\Gamma}) = Dom(\Gamma)$ and $(\Gamma \wedge \overline{\Gamma})(x) = \Gamma(x) \wedge \overline{\Gamma(x)}$ for every $x \in Dom(\Gamma)$.

13.2 Conservative extension of the simply typed lambda calculus

As a logic, $\lambda\bullet\mu$ is a conservative extension of the simply typed lambda calculus $\lambda\rightarrow$.

Definition 13.9. Syntax of λ-terms of the simply typed lambda calculus $\lambda\rightarrow$ is the same as $\lambda\bullet\mu$. Type expressions of $\lambda\rightarrow$ is defined as follows:

$$\mathbf{TExp}_{\lambda\rightarrow} \ ::= \ \mathbf{TConst} \ \mid \ \mathbf{TVar} \ \mid \ \mathbf{TExp}_{\lambda\rightarrow} \rightarrow \mathbf{TExp}_{\lambda\rightarrow}$$

Typing system $\lambda\rightarrow$ is defined by the following typing rules:

$$\dfrac{}{\Gamma \cup \{x : A\} \vdash_{\lambda\rightarrow} x : A} \ (\mathrm{var}) \qquad \dfrac{}{\Gamma \vdash_{\lambda\rightarrow} c : \tau(c)} \ (\mathrm{const})$$

$$\dfrac{\Gamma \cup \{x : A\} \vdash_{\lambda\rightarrow} M : B}{\Gamma \vdash_{\lambda\rightarrow} \lambda x.\, M : A \rightarrow B} \ (\rightarrow\mathrm{I}) \qquad \dfrac{\Gamma_1 \vdash_{\lambda\rightarrow} M : A \rightarrow B \quad \Gamma_2 \vdash_{\lambda\rightarrow} N : A}{\Gamma_1 \cup \Gamma_2 \vdash_{\lambda\rightarrow} MN : B} \ (\rightarrow\mathrm{E})$$

Definition 13.10. Let A be a type expression of $\lambda\bullet\mu$, and C a type expression of $\lambda\rightarrow$. We define a type expression $\{\!\!\{A\}\!\!\}_C$ of $\lambda\rightarrow$ as follows:

$$\begin{aligned} \{\!\!\{X\}\!\!\}_C &= X, & \{\!\!\{P\}\!\!\}_C &= P, \\ \{\!\!\{\bullet A\}\!\!\}_C &= C, & \{\!\!\{A \rightarrow B\}\!\!\}_C &= \{\!\!\{A\}\!\!\}_C \rightarrow \{\!\!\{B\}\!\!\}_C, \\ \{\!\!\{\mu X.A\}\!\!\}_C &= \{\!\!\{A\}\!\!\}_C. \end{aligned}$$

Note that $\{\!\!\{A\}\!\!\}_C = A$ if A has no occurrence of $\bullet$ or μ.

Proposition 13.11. *Let A and B be type expressions of $\lambda\bullet\mu$, and C a type expression of $\lambda\rightarrow$. If A is proper in X, then $\{\!\!\{A[B/X]\}\!\!\}_C = \{\!\!\{A\}\!\!\}_C$.*

Proof. By straightforward induction on $h(A)$. □

Proposition 13.12. *Let A and B be type expressions of $\lambda\bullet\mu$, and C a type expression of $\lambda\rightarrow$. If $A \simeq B$, then $\{\!\!\{A\}\!\!\}_C = \{\!\!\{B\}\!\!\}_C$.*

Proof. By induction on the derivation of $A \simeq B$. Use Proposition 13.11 for the cases ($\simeq$-fix) and ($\simeq$-uniq). □

Proposition 13.13. *Let A and B be type expressions of $\lambda\bullet\mu$, and C a type expression of $\lambda\rightarrow$ such that $\vdash_{\lambda\rightarrow} M : C$ is derivable for some M. Let $X_1, X_2, \ldots, X_n, Y_1, Y_2, \ldots, Y_n$ are distinct type variables. Let γ be a subtyping assumption such that $\gamma \subset \{X_1 \preceq Y_1, X_2 \preceq Y_2, \ldots, X_n \preceq Y_n\}$. If $\gamma \vdash A \preceq B$ is derivable, then $\vdash_{\lambda\rightarrow} N : \{\!\!\{A[X_1/Y_1, X_2/Y_2, \ldots, X_n/Y_n]\}\!\!\}_C \rightarrow \{\!\!\{B[X_1/Y_1, X_2/Y_2, \ldots, X_n/Y_n]\}\!\!\}_C$ is derivable for some N.*

Proof. By induction on the derivation of $\gamma \vdash A \preceq B$, and by cases of the last rule applied in the derivation. We can take M as N in the cases ($\preceq$-$\bullet$), ($\preceq$-approx) and ($\preceq$-$\bullet\rightarrow$). Let $N = \lambda x.\, x$ for ($\preceq$-$\rightarrow\bullet$). □

Theorem 13.14. *Let A a type expressions of $\lambda\bullet\mu$, and C a type expression of $\lambda\rightarrow$ such that $\vdash_{\lambda\rightarrow} M : C$ is derivable for some M. If $\Gamma \vdash N : A$ is derivable in $\lambda\bullet\mu$ for some N, then so is $\{\!\!\{\Gamma\}\!\!\}_C \vdash_{\lambda\rightarrow} L : \{\!\!\{A\}\!\!\}_C$ in $\lambda\rightarrow$ for some L, where $Dom(\{\!\!\{\Gamma\}\!\!\}_C) = Dom(\Gamma)$ and $\{\!\!\{\Gamma\}\!\!\}_C(x) = \{\!\!\{\Gamma(x)\}\!\!\}_C$ for every $x \in Dom(\Gamma)$.*

Proof. By Proposition 13.5, we can assume that the derivation of $\Gamma \vdash N : A$ includes no ($\bullet$)-rule. We prove the conjecture by induction on the derivation and by cases of the last rule applied in the derivation.

Use Proposition 13.13 in the case ($\preceq$).

Case: (*var*). Obvious because $\{\!\!\{A\}\!\!\}_C = \{\!\!\{A_i\}\!\!\}_C$ for some i ($0 \le i \le n$).

Case: (*const*). Obvious because $\{\!\!\{A\}\!\!\}_C = A$.

Case: ($\top$). Trivial because $A = \top$ and $\{\!\!\{A\}\!\!\}_C = C$ in this case.

Case: ($\preceq$). For some B', the derivation ends with:

$$\dfrac{\begin{array}{cc}\vdots & \vdots \\ \Gamma \vdash N : A' & \vdash A' \preceq A\end{array}}{\Gamma \vdash N : A}\ (\bullet)$$

By the induction hypothesis, $\{\!\!\{\Gamma\}\!\!\}_C \vdash_{\lambda\rightarrow} L : \{\!\!\{A'\}\!\!\}_C$ is derivable for some L. On the other hand, by Proposition 13.13, $\vdash_{\lambda\rightarrow} N : \{\!\!\{A'\}\!\!\}_C \rightarrow \{\!\!\{A\}\!\!\}_C$ is derivable for some N. Therefore, so is $\{\!\!\{\Gamma\}\!\!\}_C \vdash_{\lambda\rightarrow} N L : \{\!\!\{A\}\!\!\}_C$.

Case: ($\rightarrow$*I*). For some y, K, A_1 and A_2 such that $y \neq Dom(\Gamma)$, $N = \lambda y.\, K$ and $A = A_1 \rightarrow A_2$, the derivation ends with:

$$\dfrac{\begin{array}{c}\vdots \\ \Gamma \cup \{y : A_1\} \vdash N : A_2\end{array}}{\Gamma \vdash \lambda y.\, K : A_1 \rightarrow A_2}\ (\rightarrow\mathrm{I})$$

By the induction hypothesis, $\{\!\!\{\Gamma\}\!\!\}_C \cup \{y : \{\!\!\{A_1\}\!\!\}_C\} \vdash_{\lambda\rightarrow} L : \{\!\!\{A_2\}\!\!\}_C$ is derivable for some L. Therefore, so is $\{\!\!\{\Gamma\}\!\!\}_C \vdash_{\lambda\rightarrow} \lambda y.\, L : \{\!\!\{A\}\!\!\}_C$ because $\{\!\!\{A\}\!\!\}_C = \{\!\!\{A_1\}\!\!\}_C \rightarrow \{\!\!\{A_2\}\!\!\}_C$.

Case: ($\rightarrow$*E*). For some n, N_1, N_2, A' and B such that $N = N_1 N_2$ and $A = \bullet^n A'$, the derivation ends with:

$$\dfrac{\begin{array}{cc}\vdots & \vdots \\ \Gamma \vdash N_1 : \bullet^n(B \rightarrow A') & \Gamma \vdash N_2 : \bullet^n B\end{array}}{\Gamma \vdash N_1 N_2 : \bullet^n A'}\ (\rightarrow\mathrm{E})$$

By the induction hypothesis, $\{\!\!\{\Gamma\}\!\!\}_C \vdash_{\lambda\rightarrow} L_1 : \{\!\!\{\bullet^n(B \rightarrow A')\}\!\!\}_C$ and $\{\!\!\{\Gamma\}\!\!\}_C \vdash_{\lambda\rightarrow} L_2 : \{\!\!\{\bullet^n B\}\!\!\}_C$ are derivable for some L_1 and L_2. If $n = 0$, i.e., $A = A'$, then we can derive $\{\!\!\{\Gamma\}\!\!\}_C \vdash_{\lambda\rightarrow} L_1 L_2 : \{\!\!\{A\}\!\!\}_C$ because $\{\!\!\{\bullet^n(B \rightarrow A')\}\!\!\}_C = \{\!\!\{B\}\!\!\}_C \rightarrow \{\!\!\{A\}\!\!\}_C$ and $\{\!\!\{\bullet^n B\}\!\!\}_C = \{\!\!\{B\}\!\!\}_C$. On the other hand, if $n > 0$, we can obviously derive $\{\!\!\{\Gamma\}\!\!\}_C \vdash_{\lambda\rightarrow} M : \{\!\!\{A\}\!\!\}_C$ because $\{\!\!\{A\}\!\!\}_C = \{\!\!\{\bullet^n A'\}\!\!\}_C = C$. □

Corollary 13.15. *Let A a type expression of $\lambda{\to}$, i.e., A has no occurrence of $\bullet$ or μ, and Γ a typing context of $\lambda{\to}$, i.e., $\Gamma(x)$ has no occurrence of $\bullet$ or μ for every $x \in Dom(\Gamma)$. If $\Gamma \vdash M : A$ is derivable in $\lambda\bullet\mu$ for some M, then so is $\Gamma \vdash_{\lambda\to} N : A$ in $\lambda{\to}$ for some N.*

Proof. Obvious from Theorem 13.14 because $\{\!\!\{A\}\!\!\}_{X\to X} = A$, $\{\!\!\{\Gamma\}\!\!\}_{X\to X} = \Gamma$, and $\vdash_{\lambda\to} \lambda x.\, x : X \to X$ is derivable in $\lambda{\to}$. □

14 $\lambda\bullet\mu$ as a basis for logic of programs

The typing system $\lambda\bullet\mu$ and its interpretation can be easily extended to cover full propositional and second-order types. For example, we can add the following rules for product types.

$$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash N : B}{\Gamma \vdash {<}M, N{>} : A \times B}\,(\times\mathrm{I}) \qquad \frac{\Gamma \vdash M : \bullet^n(A \times B)}{\Gamma \vdash \mathbf{p_1}\, M : \bullet^n A}\,(\times\mathrm{E}) \qquad \frac{\Gamma \vdash M : \bullet^n(A \times B)}{\Gamma \vdash \mathbf{p_2}\, M : \bullet^n B}\,(\times\mathrm{E})$$

With the help of such extensions, $\lambda\bullet\mu$ can be a basis for logic of a wide range of programs. In this section we give some examples.

Streams. Streams, or infinite sequences, of data of a type X are representable by the type $\mu Y.X \times \bullet Y$. Since this type is positively finite, its elements are all maximal. We can construct recursive programs for streams with the fixed point combinators $\mathbf{Y}$; for example, a program that generates a stream of a given constant of type X as follows, where $A = \mu Y.X \times \bullet Y$.

$$\begin{prooftree}
\AxiomC{$\vdots$}
\noLine
\UnaryInfC{$\vdash \mathbf{Y} : (\bullet A \to A) \to A$}
\AxiomC{}
\RightLabel{(var)}
\UnaryInfC{$x : X,\, y : \bullet A \vdash x : X$}
\AxiomC{}
\RightLabel{(var)}
\UnaryInfC{$x : X,\, y : \bullet A \vdash y : \bullet A$}
\RightLabel{$(\times\mathrm{I})$}
\BinaryInfC{$x : X,\, y : \bullet A \vdash {<}x, y{>} : X \times \bullet A$}
\RightLabel{$(\simeq)$}
\UnaryInfC{$x : X,\, y : \bullet A \vdash {<}x, y{>} : A$}
\RightLabel{$(\to\mathrm{I})$}
\UnaryInfC{$x : X \vdash \lambda y.\, {<}x, y{>} : \bullet A \to A$}
\RightLabel{$(\to\mathrm{E})$}
\BinaryInfC{$x : X \vdash \mathbf{Y}(\lambda y.\, {<}x, y{>}) : A$}
\RightLabel{$(\to\mathrm{I})$}
\UnaryInfC{$\vdash \lambda x.\, \mathbf{Y}(\lambda y.\, {<}x, y{>}) : X \to A$}
\end{prooftree}$$

The next example shows the derivation of the program $\mathbf{Y}(\lambda f.\, \lambda x.\, \lambda y.\, {<}\mathbf{p_1}\, x, f y (\mathbf{p_2}\, x){>})$, which merges two streams. Let $B = (\bullet(A \to A \to A) \to (A \to A \to A)) \to (A \to A \to A)$ and $\Gamma = \{f : \bullet(A \to A \to A),\ x : A,\ y : A\}$.

$$\begin{prooftree}
\AxiomC{$\vdots$}
\noLine
\UnaryInfC{$\vdash \mathbf{Y} : B$}
\AxiomC{}
\RightLabel{(var)}
\UnaryInfC{$\Gamma \vdash x : A$}
\RightLabel{$(\simeq)$}
\UnaryInfC{$\Gamma \vdash x : X \times \bullet A$}
\RightLabel{$(\times\mathrm{E})$}
\UnaryInfC{$\Gamma \vdash \mathbf{p_1}\, x : X$}
\AxiomC{}
\RightLabel{(var)}
\UnaryInfC{$\Gamma \vdash f : \bullet(A \to A \to A)$}
\AxiomC{}
\RightLabel{(var)}
\UnaryInfC{$\Gamma \vdash y : A$}
\RightLabel{$(\preceq)$}
\UnaryInfC{$\Gamma \vdash y : \bullet A$}
\RightLabel{$(\to\mathrm{E})$}
\BinaryInfC{$\Gamma \vdash f y : \bullet(A \to A)$}
\AxiomC{}
\RightLabel{(var)}
\UnaryInfC{$\Gamma \vdash x : A$}
\RightLabel{$(\simeq)$}
\UnaryInfC{$\Gamma \vdash x : X \times \bullet A$}
\RightLabel{$(\times\mathrm{E})$}
\UnaryInfC{$\Gamma \vdash \mathbf{p_2}\, x : \bullet A$}
\RightLabel{$(\to\mathrm{E})$}
\BinaryInfC{$\Gamma \vdash f y (\mathbf{p_2}\, x) : \bullet A$}
\RightLabel{$(\times\mathrm{I})$}
\BinaryInfC{$\Gamma \vdash {<}\mathbf{p_1}\, x, f y (\mathbf{p_2}\, x){>} : X \times \bullet A$}
\RightLabel{$(\simeq)$}
\UnaryInfC{$\Gamma \vdash {<}\mathbf{p_1}\, x, f y (\mathbf{p_2}\, x){>} : A$}
\RightLabel{$(\to\mathrm{I})$}
\UnaryInfC{$f : \bullet(A \to A \to A),\, x : A \vdash \lambda y.\, {<}\mathbf{p_1}\, x, f y (\mathbf{p_2}\, x){>} : A \to A$}
\RightLabel{$(\to\mathrm{I})$}
\UnaryInfC{$f : \bullet(A \to A \to A) \vdash \lambda x.\, \lambda y.\, {<}\mathbf{p_1}\, x, f y (\mathbf{p_2}\, x){>} : A \to A \to A$}
\RightLabel{$(\to\mathrm{I})$}
\UnaryInfC{$\vdash \lambda f.\, \lambda x.\, \lambda y.\, {<}\mathbf{p_1}\, x, f y (\mathbf{p_2}\, x){>} : \bullet(A \to A \to A) \to A \to A \to A$}
\RightLabel{$(\to\mathrm{E})$}
\BinaryInfC{$\vdash \mathbf{Y}\, (\lambda f.\, \lambda x.\, \lambda y.\, {<}\mathbf{p_1}\, x, f y (\mathbf{p_2}\, x){>}) : A \to A \to A$}
\end{prooftree}$$

More complicated programs over streams such as the prime number generator based on the sieve of Eratosthenes are also derivable by extending our logic to a predicate logic endowed with an arithmetic (powerful enough to handle, e.g., primitive recursive functions) for annotation, and by allowing type expressions such as $\bullet^t A$ as well-formed type expressions, with t being a numeric expression.

McCarthy's 91-function. Provided such an extension to predicate logic, we can construct a wide range of recursive programs with fixed point combinators assuring their termination. Consider the following recursive program, which represents McCarthy's 91-function:

$$f\ x \equiv \textbf{if}\ (x > 100)\ \textbf{then}\ x - 10\ \textbf{else}\ f\ (f\ (x + 11)).$$

We can show that f has a type, or satisfies a specification, $\forall n.\, \textbf{nat}(n) \to \bullet^{101 \dot{-} n}\textbf{nat}(g(n))$, where n ranges over non-negative integers, and $\textbf{nat}(n)$ represents the implementation of the non-negative integer n; $\dot{-}$ and g are primitive recursive functions defined in the arithmetic as:

$$x \dot{-} y \equiv \begin{cases} x - y & \text{(if } x \geq y\text{)} \\ 0 & \text{(otherwise)} \end{cases}$$
$$g(x) \equiv \begin{cases} x - 10 & \text{(if } x > 100\text{)} \\ 91 & \text{(otherwise)} \end{cases}$$

and $\forall n.\, A(n)$ is interpreted as:

$$\mathcal{I}(\forall n.\, A(n))^{\xi}_{k} = \{\ u \mid u \in \mathcal{I}(A(n))^{\xi}_{k}\ \text{ for all } n\ \}.$$

Suppose that $f : \bullet\, \forall n.\, \textbf{nat}(n) \to \bullet^{101 \dot{-} n}\textbf{nat}(g(n))$ and $x : \textbf{nat}(n)$. The type of $\mathbf{Y}$ assures that it suffices to show:

$$\textbf{if}\ (x > 100)\ \textbf{then}\ x - 10\ \textbf{else}\ f\ (f\ (x + 11)) \qquad : \forall n.\, \textbf{nat}(n) \to \bullet^{101 \dot{-} n}\textbf{nat}(g(n)).$$

We assume that $-$ and $+$ satisfy $\forall m.\, \forall n.\, \textbf{nat}(m) \to \textbf{nat}(n) \to \textbf{nat}(n \dot{-} m)$ and $\forall m.\, \forall n.\, \textbf{nat}(m) \to \textbf{nat}(n) \to \textbf{nat}(n{+}m)$, respectively. First, we get:

$$f\ (x{+}11) : \bullet\bullet^{101 \dot{-} (n+11)}\textbf{nat}(g(n{+}11)).$$

If $n \leq 90$, then we get $f\ (x{+}11) : \bullet^{91 \dot{-} n}\ \textbf{nat}(91)$ by the definitions of $\dot{-}$ and g; and therefore, $f\ (f\ (x{+}11)) : \bullet^{91 \dot{-} n}\ \bullet^{101 \dot{-} 91}\textbf{nat}(g(91))$, which is equivalent to $\bullet^{101 \dot{-} n}\textbf{nat}(91)$. On the other hand, if $90 < n \leq 100$, then we similarly get $f\ (x{+}11) : \bullet\, \textbf{nat}(n{+}1)$; and therefore, $f\ (f\ (x{+}11)) : \bullet\bullet^{101 \dot{-} (n+1)}\textbf{nat}(g(n{+}1))$, which is also equivalent to $\bullet^{101 \dot{-} n}\textbf{nat}(91)$. Otherwise, i.e., if $100 < n$, obviously $x{-}10 : \textbf{nat}(g(n))$.

In this derivation, the fixed point combinator worked as the induction scheme discussed in Introduction with a sequence $S_0, S_1, S_2, \ldots, S_n$ as follows:

$$S_0 = \mathcal{V}$$
$$S_{k+1} = \{\ f \mid \forall n \geq 101 \dot{-} k.\ f(n) = g(n)\ \}.$$

Note also that if $\mathcal{I}(\textbf{nat}(n))^{\xi}_{k}$ does not depend on k, then the interpretation of $\vdash f : \forall n.\, \textbf{nat}(n) \to \bullet^{101 \dot{-} n}\textbf{nat}(g(n))$ implies $f \in \mathcal{I}(\forall n.\, \textbf{nat}(n) \to \textbf{nat}(g(n)))^{\xi}_{k}$ for every k. Therefore, the apparent complexity of the type expression is not essential, and it can be observed that $\vdash f : \forall n.\, \textbf{nat}(n) \to \textbf{nat}(g(n))$ becomes formally derivable from $\vdash f : \forall n.\, \textbf{nat}(n) \to \bullet^{101 \dot{-} n}\textbf{nat}(g(n))$ if we introduce another modality, say $\Box$, which is interpreted as:

$$\mathcal{I}(\Box A)^{\xi}_{k} = \{\ u \mid u \in \mathcal{I}(A)^{\xi}_{l}\ \text{for every } l\ \},$$

and accordingly enjoys the following subtyping relations and typing rules:

- $A \preceq B$ implies $\Box A \preceq \Box B$
- $\Box(A \to B) \preceq \Box A \to \Box B$
- $\Box A \preceq A$
- $\Box A \preceq \Box\Box A$
- $\Box\bullet^{t} A \preceq \Box A$
- $\textbf{nat}(n) \preceq \Box\textbf{nat}(n)$

$$\frac{\Gamma\ \vdash\ M : A}{\Box\Gamma\ \vdash\ M : \Box A}\ (\Box) \qquad \frac{\Box\Gamma_1 \cup \bullet\Gamma_2\ \vdash\ M : \bullet A}{\Box\Gamma_1 \cup \Gamma_2\ \vdash\ M : A}\ (\bullet)$$

The $(\bullet)$-rule supersedes the original one. Recursive type variables are not allowed to occur in scopes of the $\Box$-operator, and $\mathcal{I}(A)^{\xi}_{k}$ is now defined by induction on the lexicographic ordering of $<b(A),\ k,\ r(A)>$, where $b(A)$ is the depth of nesting occurrences of $\Box$ in A.

The **Nat**(n)*-example.* We now reconsider the example of object-oriented natural numbers with an addition method. We revise the definition of **Nat**(n) as follows:

$$\begin{aligned}\mathbf{Nat}(n) \equiv\ & ((n=0) + (n>0 \wedge \bullet\,\mathbf{Nat}(n{-}1)) \\ & \times (\forall m.\, \bullet\,\mathbf{Nat}(m) \to \bullet\,\mathbf{Nat}(n{+}m))).\end{aligned}$$

Then, the specifications of **add** and **add**$'$ are now different as follows:

$$\begin{aligned}\mathbf{add} &: \forall n.\, \forall m.\, \mathbf{Nat}(n) \to \bullet\,\mathbf{Nat}(m) \to \bullet\,\mathbf{Nat}(n{+}m) \\ \mathbf{add}' &: \forall n.\, \forall m.\, \bullet\,\mathbf{Nat}(n) \to \mathbf{Nat}(m) \to \bullet\,\mathbf{Nat}(n{+}m)\end{aligned}$$

We can show $\mathbf{s} : \forall n.\, \mathbf{Nat}(n) \to \mathbf{Nat}(n{+}1)$ by deriving $<\mathbf{i_2}\, x, \lambda y.\, \mathbf{add}\; x\; (\mathbf{s}\; y)> : \mathbf{Nat}(n{+}1)$ from $\mathbf{s} : \bullet\,\forall n.\, \mathbf{Nat}(n) \to \mathbf{Nat}(n{+}1)$ and $x : \mathbf{Nat}(n)$. Obviously,

$$\mathbf{i_2}\, x : (n{+}1 = 0) + (n{+}1 > 0 \wedge \bullet\,\mathbf{Nat}(n{+}1{-}1)).$$

If $y : \bullet\,\mathbf{Nat}(m)$, we get $\mathbf{s}\; y : \bullet\,\mathbf{Nat}(m{+}1)$, and consequently,

$$\mathbf{add}\; x\; (\mathbf{s}\; y) : \bullet\mathbf{Nat}(n{+}1{+}m).$$

We thus get $<\mathbf{i_2}\, x, \lambda y.\, \mathbf{add}\; x\; (\mathbf{s}\; y)> : \mathbf{Nat}(n{+}1)$. Note that, on the other hand, under similar assumptions, we can only get

$$\mathbf{add}'\; x\; (\mathbf{s}'\; y) : \bullet\bullet\,\mathbf{Nat}(n{+}1{+}m),$$

and fail to derive $\mathbf{s}' : \forall n.\, \mathbf{Nat}(n) \to \mathbf{Nat}(n{+}1)$.

15 Concluding Remarks

We have presented a modal typing system with recursive types and shown its soundness with respect to a realizability interpretation and the convergence of well-typed terms according to their types. The decidability questions for type checking, typability and inhabitation of $\lambda\bullet\mu$ types are still open. Although we presented it as a typing system, we do not intend to apply it directly to type systems of programming languages. Since our framework asserts the convergence of derived programs, typing general recursive programs naturally requires some (classical) arithmetics as seen in the case of the 91-function, which would make mechanical type checking impossible. Our goal is to capture a wider range of programs in the proofs-as-programs paradigm and give an axiomatic semantics to them preserving the compositionality of programs. We have seen that our approach is applicable to some interesting programs such as fixed point combinators and objects with binary methods, which have not been captured in the conventional frameworks.

Similar results concerning the existence of fixed points of proper type expressions (Lemma 10.3.1 in our case) could historically go back to the fixed point theorem of the logic of provability (see [5, 18]). The difference is that our logic is intuitionistic, and fixed points are treated as sets of realizers. Interestingly, by applying $(\preceq)$ to the type $(\bullet X \to X) \to X$ of the fixed point combinators, we can also derive $\bullet(\bullet X \to X) \to \bullet X$, i.e., Löb's axiom schema $\Box(\Box\phi \to \phi) \to \Box\phi$ representing the well-foundedness of the (classical) Kripke frame. It should be observed that $(\Box\phi \to \phi) \to \phi$ is valid based on intuitionistic frames, where $\phi \to \Box\phi$ is valid, if and only if the frame is well-founded.

Our intended semantics of $\lambda\bullet\mu$ suggests that our modal logic could be related to some temporal logic of discrete, linear time with finite past and infinite future, where the modal operator $\bullet$ corresponds to the "previous time", or "yesterday", modality. Gabbay and Hodkinson discussed such a temporal logic and its fixed point operator [12, 16]; however, the relationship with $\lambda\bullet\mu$ is not obvious since their temporal logic is based on classical logic.

Acknowledgments

The author is greatly indebted to Professor Solomon Feferman for the opportunity to develop the main part of this research in a stimulating environment at Stanford University.

References

1. M. Abadi and L. Cardelli. *A theory of objects*. Springer-Verlag, 1996.
2. R. M. Amadio. Recursion over realizability structure. *Information and Computation*, 91(1):55–85, 1991.
3. R. M. Amadio and L. Cardelli. Subtyping recursive types. *ACM Transactions on Programming Languages and Systems*, 15(4):575–631, 1993.
4. H. P. Barendregt. Lambda calculi with types. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2, pages 118–309. Oxford University Press, 1992.
5. G. Boolos. *The logic of provability*. Cambridge University Press, 1993.
6. K. B. Bruce, L. Cardelli, and B. C. Pierce. Comparing object encodings. *Information and Computation*, 155:108–133, 1999.
7. L. Cardelli. Amber. In G. Cousineau, P.-L. Curien, and B. Robinet, editors, *Combinators and functional programming languages*, volume 242 of *Lecture Notes in Computer Science*, pages 21–47. Springer-Verlag, 1986.
8. F. Cardone and M. Coppo. Type inference with recursive types: syntax and semantics. *Information and Computation*, 92(1):48–80, 1991.
9. R. L. Constable, S. Allen, H. Bromely, W. Cleveland, et al. *Implementing Mathematics with the Nuprl Proof Development System*. Prentice-Hall, 1986.
10. R. L. Constable and N. P. Mendler. Recursive definitions in type theory. In *Logics of Programs*, volume 193 of *Lecture Notes in Computer Science*, pages 61–78. Springer-Verlag, 1985.
11. R. L. Constable and S. F. Smith. Partial objects in constructive type theory. In *Proceedings of the 2nd IEEE Symposium on Logic in Computer Science*, pages 183–193. IEEE Computer Society Press, 1987.
12. D. M. Gabbay. The declarative past and imperative future. In *Temporal logic in specification*, volume 398 of *Lecture Notes in Computer Science*, pages 409–448. Springer-Verlag, 1989.
13. Y. Gurevich and S. Shelah. Fixed-point extensions of first-order logic. *Annals of Pure and Applied Logic*, 32(3):265–280, 1986.
14. S. Hayashi and H. Nakano. *PX: A Computational Logic*. The MIT Press, 1988.
15. R. Hindley. The completeness theorem for typing λ-terms. *Theoretical Computer Science*, 22:1–17, 1983.
16. I. M. Hodkinson. On Gabbay's temporal fixed point operator. *Theoretical Computer Science*, 139:1–25, 1995.
17. W. A. Howard. The formulae-as-types notion of construction. In R. J. Hindley and J. P. Seldin, editors, *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 480–490. Academic Press, 1980.
18. G. Japaridze and D. de Jongh. The logic of provability. In *Handbook of proof theory*, pages 475–546. North Holland, 1998.
19. S. Kobayashi and M. Tatsuta. Realizability interpretation of generalized inductive definitions. *Theoretical Computer Science*, 131(1):121–138, 1994.
20. D. C. Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27(3):333–354, 1983.
21. D. Leivant. Typing and computational properties of lambda expressions. *Theoretical Computer Science*, 44(1):51–68, 1986.
22. D. B. MacQueen, G. D. Plotkin, and R. Sethi. An ideal model for recursive polymorphic types. *Information and Computation*, 71:95–130, 1986.
23. H. Nakano. A modality for recursion. In *Proceedings of the 15th IEEE Symposium on Logic in Computer Science*, pages 255–266. IEEE Computer Society Press, 2000.
24. C. Paulin-Mohring. Extracting F_ω's programs from proofs in the calculus of constructions. In *Proceedings of the 16th ACM Symposium on Principles of Programming Languages*, pages 89–104, 1989.
25. V. R. Pratt. A decidable μ-calculus (preliminary report). In *Proceedings of the 22nd IEEE Symposium on Foundation of Computer Science*, pages 421–427, 1981.
26. D. S. Scott. A type-theoretical alternative to ISWIM, CUCH, OWHY. *Theoretical Computer Science*, 121:411–440, 1993.
27. M. Tatsuta. Realizability interpretation of coinductive definitions and program synthesis with streams. *Theoretical Computer Science*, 122:119–136, 1994.