Conjunctive types and useless-code elimination

Ferruccio Damiani
Dipartimento di Informatica, Università di Torino,
Corso Svizzera 185, 10149 Torino, Italy
Tel. (+39) 011 6706719, Fax. (+39) 011 751603, E-mail: damiani@di.unito.it

Abstract

We investigate the use of *conjunctive non-standard type inference* for the elimination of useless-code in higher-order typed functional programs. Useless-code elimination can be performed by optimizing compilers to improve the quality of the code produced. This kind of optimization is particularly useful when dealing with programs obtained by automatic translation from constructive proofs.

Key words: non-standard type inference, program analysis, useless-code.

1 Introduction

Useless-code elimination is a program transformation technique that eliminates subexpression that does not contribute to the final result of a program. Useless-code elimination focused on the elimination of useless function's parameters is often called *useless-variable elimination*. Various techniques for removing useless-code in higher-order functional programs have been proposed (see, for instance, [17, 1, 2, 3, 7, 6, 16, 19, 11, 9]). Useless-code elimination is particularly useful when dealing with programs obtained by automatic translation from constructive proofs (see [13, 12] for an introduction to the subject), and the use of types is at the heart of the techniques developed in this context.

The basic idea for detecting useless-code via a typing system is to use type inference rules to decorate terms with *non-standard types* saying that some subterms are useless-code while others may be useful to compute the value of the term. Take for instance the language PCF [15] with the primitive type of natural numbers, nat. We will consider two non-standard types for PCF terms of type nat:

δ^{nat} : saying that the term *matters* (it *may be used*), and
ω^{nat} : saying that the term *is not used*.

For instance, given the term $P = (\lambda x^{\mathsf{nat}}.3)y^{\mathsf{nat}}$ it is easy to see that the decoration

$$P' = ((\lambda x.3)y^{\omega^{\mathsf{nat}}})^{\delta^{\mathsf{nat}}} \quad , \tag{1}$$

which states that the term P may be used while its subterm y^{nat} is not used, is correct[1]. Another decoration of P is $P'' = ((\lambda x.3)y^{\delta^{\mathsf{nat}}})^{\delta^{\mathsf{nat}}}$ which, although correct, is

[1] Under a *call-by-name* evaluation strategy.

"less informative" than P', since it says that every subterm of P may be used. Given a non-standard type assignment system for useless-code analysis $\vdash$ and a term M, we call *optimal $\vdash$-typing of M* any decoration of M by $\vdash$ that shows all the useless-code of M that can be detected by using $\vdash$. Note that the decoration (1) shows all the useless-code of P, so it is optimal w.r.t. all the systems in which it can be derived.

Take a PCF program

$$M = (\lambda f^{\rho}.U)V \ ,$$

and suppose that the variable f^{ρ} in U has two different uses and two different non-standard types ϕ_1 and ϕ_2 (corresponding to that uses). Which non-standard type should we assign to V? We can consider three different solutions.

1. Considering only those decorations in which ϕ_1 and ϕ_2 are equal. This solution (which is essentially the one adopted in [1, 3, 11, 9]) results in a useless-variable elimination and may prevent us to discover some useless-code.

2. Considering only those decorations in which both ϕ_1 and ϕ_2 are entailed by a more informative non-standard type, say ϕ, that can be assigned to V. Also this solution (which is essentially the one adopted in [2, 4, 7, 6]) may prevent us to discover some useless-code.

3. Assign both ϕ_1 and ϕ_2 to V, i.e., assign to V the *conjunction*, $\phi_1 \wedge \phi_2$, of the two non-standard types. In this way there is no restriction on the decoration that may prevent us to discover the useless-code exploited by the two different non-standard types of V.

The aim of this paper is to investigate the third solution, that is the use of *conjunctive non-standard type inference* for the elimination of useless-code in functional programs.

In Section 2 of this paper we introduce the programming language we are dealing with and its operational semantics. In Section 3 we briefly show how program properties can be represented by partial equivalence relations on a term model of the programming language. In Section 4 we describe the set of non-standard types for useless-code analysis and its semantics, we also give an axiomatization of the logical implication (entailment) between non-standard types. In Section 5 we present the non-standard type assignment system and a mapping that simplifies a program by removing all the useless-code that can be detected by using the system.

A preliminary version of the material presented in this paper appeared in Chapter 8 of [5], and a restricted version of the system, in which conjunction is limited to rank 2 and not used to deal with recursive programs, has been presented in [8] (see also Chapter 9 of [5]). The main result given in the present paper (existence of optimal simplifications) is new.

2 The language PCFP

In this section we introduce the functional programming language PCFP ("Programming Computable Functions with Pairs") and its operational semantics. PCFP is the dialect of the language PCF [15] obtained by using natural numbers instead of integers and adding a type constructor for pairs.

Definition 2.1 (PCFP types) *The only* ground *type considered is the set of natural numbers,* nat. *The set of* types *(*$\mathbf{T}$*), ranged over by* ρ, σ, τ, *is defined by the following grammar:* $\rho ::= \mathsf{nat} \mid \rho \to \rho \mid \rho \times \rho$.

PCFP terms are defined from a set $\mathcal{V}$ of typed *term variables* (ranged over by $x^\rho, y^\sigma, \ldots$) and a set $\mathcal{K} = \{\mathsf{0}^{\mathsf{nat}}, \mathsf{1}^{\mathsf{nat}}, \ldots + ^{\mathsf{nat}\times\mathsf{nat}\to\mathsf{nat}}, -^{\mathsf{nat}\times\mathsf{nat}\to\mathsf{nat}}\}$ of typed *term constants* (ranged over by k). The type of a constant k is denoted by $\mathbf{T}(\mathsf{k})$.

Definition 2.2 (PCFP terms) *The set of* PCFP terms $\Lambda_{\mathbf{T}}$ *(ranged over by* M, N, $\ldots$*) is defined by:* $\Lambda_{\mathbf{T}} = \cup_{\rho\in\mathbf{T}}\, \Lambda_\rho$, *where* Λ_ρ *is the set of the* PCFP terms of type ρ, *defined by the rules in Fig. 1.*

(Var) $x^\rho \in \Lambda_\rho$ (Con) $\mathsf{k}^\rho \in \Lambda_\rho$

(→ I) $\dfrac{M \in \Lambda_\sigma}{\lambda x^\rho.M \in \Lambda_{\rho\to\sigma}}$ (→ E) $\dfrac{M \in \Lambda_{\rho\to\sigma} \quad N \in \Lambda_\rho}{MN \in \Lambda_\sigma}$

(×I) $\dfrac{M_1 \in \Lambda_{\rho_1} \quad M_2 \in \Lambda_{\rho_2}}{\langle M_1, M_2\rangle \in \Lambda_{\rho_1\times\rho_2}}$ (×E$_i$) $\dfrac{M \in \Lambda_{\rho_1\times\rho_2}}{\mathsf{prj}_i M \in \Lambda_{\rho_i}}\ i \in \{1,2\}$

(Fix) $\dfrac{M \in \Lambda_\rho}{\mathsf{fix}\, x^\rho.M \in \Lambda_\rho}$ (Ifz) $\dfrac{N \in \Lambda_{\mathsf{nat}} \quad M_1 \in \Lambda_\rho \quad M_2 \in \Lambda_\rho}{\mathsf{ifz}\, N\, \mathsf{then}\, M_1\, \mathsf{else}\, M_2 \in \Lambda_\rho}$

Figure 1: Rules for PCFP term formation

In a PCFP term M the types of variables and constants are explicitly mentioned. In the following we will often omit to write types which are clear from the context. The finite set of the *free variables* of a term M, denoted by $\mathrm{FV}(M)$, is defined in the usual way. We will identify terms modulo renaming of the bound variables. A substitution is a finite function mapping term variables to terms, denoted by $[x_1 := N_1, \ldots, x_n := N_n]$ ($[\vec{x} := \vec{N}]$ for short), which respects the types, i.e., each $x_i^{\rho_i}$ is substituted by a term N_i of the same type. Substitution acts on free variables, the renaming of the bound variables is implicitly supposed.

Let $\Lambda^{\mathbf{c}}_{\mathbf{T}}$ be the set of the *closed terms*, i.e., $\Lambda^{\mathbf{c}}_{\mathbf{T}} = \{M \in \Lambda_{\mathbf{T}} \mid \mathrm{FV}(M) = \emptyset\}$. The operational semantics of PCFP is given by an inductively defined *evaluation relation*, $M \Downarrow K$, where M is a closed term and K is a closed term in *weak head normal form* (*w.h.n.f.*), i.e. an element of the set of *values*: $\mathbf{V}_{\mathbf{T}} = \mathcal{K} \cup \{\lambda x^\rho.N \mid \lambda x^\rho.N \in \Lambda^{\mathbf{c}}_{\mathbf{T}}\} \cup \{\langle M_1, M_2\rangle \mid \langle M_1, M_2\rangle \in \Lambda^{\mathbf{c}}_{\mathbf{T}}\}$. Any PCFP constant has either type nat or $\mathsf{nat} \times \mathsf{nat} \to \mathsf{nat}$. The meaning of a functional constant k can be given by a set $\mathbf{mean}(\mathsf{k})$ of pairs, i.e. if $(K, \mathsf{k}_1) \in \mathbf{mean}(\mathsf{k})$ then $\mathsf{k}K$ evaluates to k_1. For example $(\langle 1, 3\rangle, 4) \in \mathbf{mean}(+)$.

Definition 2.3 (Value of a term) *Let* $M \in \Lambda^{\mathbf{c}}_{\mathbf{T}}$. *We write* $M \Downarrow K$, *and say that* M *evaluates to* K, *if this statement is derivable by using the rules in Fig. 2.*

Let $M \Downarrow$, to be read "M is convergent", mean that, for some K, $M \Downarrow K$, and let $M \Uparrow$, to be read "M is divergent", mean that, for no K, $M \Downarrow K$.

$$\text{(CAN)}\ \frac{K \in \mathbf{V_T}}{K \Downarrow K} \qquad \text{(APP)}\ \frac{M \Downarrow \lambda x.P \quad P[x := N] \Downarrow K}{MN \Downarrow K}$$

$$(\mathcal{K}\text{APP})\ \frac{M \Downarrow \mathsf{k} \quad N \Downarrow \langle N_1, N_2\rangle \quad N_1 \Downarrow \mathsf{k}_1 \quad N_2 \Downarrow \mathsf{k}_2}{MN \Downarrow \mathsf{k}_3}\ (\langle \mathsf{k}_1, \mathsf{k}_2\rangle, \mathsf{k}_3) \in \mathbf{mean}(\mathsf{k})$$

$$(\text{PRJ}_i)\ \frac{P \Downarrow \langle M_1, M_2\rangle \quad M_i \Downarrow K}{\mathsf{prj}_i P \Downarrow K}\ i \in \{1,2\} \qquad \text{(FIX)}\ \frac{M[x := \mathsf{fix}\, x.M] \Downarrow K}{\mathsf{fix}\, x.M \Downarrow K}$$

$$(\text{IFZ}_1)\ \frac{N \Downarrow \mathsf{0} \quad M_1 \Downarrow K}{\mathsf{ifz}\, N\, \mathsf{then}\, M_1\, \mathsf{else}\, M_2 \Downarrow K} \qquad (\text{IFZ}_2)\ \frac{N \Downarrow \mathsf{n} \quad M_2 \Downarrow K}{\mathsf{ifz}\, N\, \mathsf{then}\, M_1\, \mathsf{else}\, M_2 \Downarrow K}\ \mathsf{n} \neq \mathsf{0}$$

Figure 2: "Natural semantics" evaluation rules

Following [14] we introduce the *ground contextual equivalence* on PCFP terms, which is the congruence on terms induced by the contextual preorder that compares the termination behaviour of programs just at the ground type nat. This amounts to assume that *complete PCFP programs* are closed terms of type nat, and that the only observable behaviour of a complete program P is its divergence or converge to some natural number. Let $(\mathcal{C}[\ ]^\rho)^\sigma$ denote a typed context of type σ with a hole of type ρ in it.

Definition 2.4 (Ground contextual equivalence) *Let M and N be terms of type ρ. Define $M \preceq_{\text{obs}} N$ whenever, for all closed contexts $(\mathcal{C}[\ ]^\rho)^{\mathsf{nat}}$, if $\mathcal{C}[M]$ and $\mathcal{C}[N]$ are closed terms, then $\mathcal{C}[M] \Downarrow$ implies $\mathcal{C}[N] \Downarrow$. The relation $\preceq_{\text{obs}}$ is the* ground contextual preorder *and the equivalence induced by $\preceq_{\text{obs}}$, denoted by $\simeq_{\text{obs}}$, is the* ground observational equivalence.

The *closed term model* $\mathcal{M}$ of PCFP is defined by interpreting each type ρ as the set of the equivalence classes of the relation $\simeq_{\text{obs}}$ on the closed terms of type ρ in $\Lambda^{\mathsf{c}}_{\mathbf{T}}$. Let $[M]$ denote the equivalence class of the closed term M, and let the extension of the the relation $\preceq_{\text{obs}}$ to equivalence classes be defined by: $[M] \preceq_{\text{obs}} [N]$ if and and only if $M \preceq_{\text{obs}} N$. Then, the interpretation of type ρ in the model $\mathcal{M}$, $\mathbf{I}(\rho)$, is a poset w.r.t. $\preceq_{\text{obs}}$. For every type ρ, $[\mathsf{fix}\, x^\rho.x]$ (the equivalence class of the divergent computation of type ρ), is the least element of $\mathbf{I}(\rho)$. An *environment* is a mapping $e : \mathcal{V} \to \bigcup_{\rho \in \mathbf{T}} \mathbf{I}(\rho)$ which respects types, i.e. a mapping such that, for all x^ρ, $e(x^\rho) \in \mathbf{I}(\rho)$. The interpretation of a term M in an environment e is defined by: $[\![M]\!]_e = [M[x_1 := N_1, \ldots, x_n := N_n]]$, where $\{x_1, \ldots, x_n\} = \text{FV}(M)$ and $[N_l] = e(x_l)$ $(1 \leq l \leq n)$. Note that $\mathcal{M}$ is an extensional model.

3 P.e.r.s over $\mathcal{M}$ as program properties

A *partial equivalence relation* (*p.e.r.* for short) over a set A is a symmetric and transitive binary relation over A. Let $\mathcal{R}$ be a p.e.r. over a set A, we call *domain of* $\mathcal{R}$ the subset of A defined as: $\text{Dom}(\mathcal{R}) = \{E \mid (E,E) \in \mathcal{R}\}$. Note that $\mathcal{R}$ is reflexive on its domain. The use of p.e.r.'s for describing program analysis is well known in literature and dates back at least to the proposal in [10].

In the following let "p.e.r. over ρ" mean "p.e.r. over $\mathbf{I}(\rho)$" ($\mathbf{I}(\rho)$ is the interpretation of the type ρ in the closed term model $\mathcal{M}$). In this section we explain how some p.e.r.'s over ρ can be read as properties of PCFP programs of type ρ.

The following definition formally explains what is meant by "a program P of type ρ satisfies the property (p.e.r.) $\mathcal{R}$ over ρ".

Definition 3.1 (P satisfies $\mathcal{R}$) *Let $\mathcal{R}$, $\mathcal{R}_1, \ldots, \mathcal{R}_n$ ($n \geq 0$) be p.e.r. over types ρ, $\rho_1, \ldots, \rho_n$, respectively. We say that a program P of type ρ with free variables $x_1^{\rho_1} \ldots, x_n^{\rho_n}$ satisfies the property $\mathcal{R}$ under the assumptions $\mathcal{R}_i$ for $x_i^{\rho_i}$ ($1 \leq i \leq n$) if, for all the environments e and e' such that $(e(x_i^{\rho_i}), e'(x_i^{\rho_i})) \in \mathcal{R}_i$, we have that $(\llbracket M \rrbracket_e, \llbracket M \rrbracket_{e'}) \in \mathcal{R}$.*

Given a p.e.r. $\mathcal{R}$ over ρ, to understand which is the property represented, it is useful to observe that $\mathcal{R}$ can be read as a relation over the closed terms of type ρ: $M, N \in \Lambda^{\mathbf{c}}_{\mathbf{T}}$ are related by $\mathcal{R}$ if and only if $([M], [N]) \in \mathcal{R}$.

- For every type $\rho \in \mathbf{T}$, the diagonal p.e.r. over ρ: $\Delta^\rho = \{([M], [M]) \mid [M] \in \mathbf{I}(\rho)\}$, which equates each element of $\mathbf{I}(\rho)$ with only itself, is the property for which the value of a term *does matter*.

- For every type $\rho \in \mathbf{T}$, the trivial p.e.r. over ρ: $\Omega^\rho = \{([M], [N]) \mid [M], [N] \in \mathbf{I}(\rho)\}$, which equates all the elements of $\mathbf{I}(\rho)$, is the "true" property, for which the value of a term *does not matter*.

- Given a p.e.r. $\mathcal{R}_1$ over ρ_1 and a p.e.r. $\mathcal{R}_2$ over ρ_2, let $\mathcal{R}_1 \twoheadrightarrow \mathcal{R}_2$ be the p.e.r. over $\rho_1 \to \rho_2$ defines as: $\mathcal{R}_1 \twoheadrightarrow \mathcal{R}_2 = \{([F], [G]) \mid \forall ([M], [N]) \in \mathcal{R}_1.([FM], [GN]) \in \mathcal{R}_2\}$. For instance $\Omega^{\mathsf{nat}} \to \Delta^{\mathsf{nat}}$ is the property (satisfied by all the, necessary constant, functions of type $\mathsf{nat} \to \mathsf{nat}$ which do not use their argument) which says that the value of the application of the function can be used without using the argument. Note that, for every $\rho_1, \rho_2 \in \mathbf{T}$, $\Delta^{\rho_1} \twoheadrightarrow \Delta^{\rho_2} = \Delta^{\rho_1 \to \rho_2}$ and $\Omega^{\rho_1} \twoheadrightarrow \Omega^{\rho_2} = \Omega^{\rho_1 \to \rho_2} = \mathcal{R}_1 \twoheadrightarrow \Omega^{\rho_2}$ (for any p.e.r. $\mathcal{R}_1$ over ρ_1).

- Given a p.e.r. $\mathcal{R}_1$ over ρ_1 and a p.e.r. $\mathcal{R}_2$ over ρ_2, let $\mathcal{R}_1 \divideontimes \mathcal{R}_2$ be the p.e.r. over $\rho_1 \times \rho_2$ defined as: $\mathcal{R}_1 \divideontimes \mathcal{R}_2 = \{([M], [N]) \mid \forall i \in \{1, 2\}.([\mathsf{prj}_i M], [\mathsf{prj}_i N]) \in \mathcal{R}_i\}$. For every $\rho_1, \rho_2 \in \mathbf{T}$, $\Delta^{\rho_1} \divideontimes \Delta^{\rho_2} = \Delta^{\rho_1 \times \rho_2}$ and $\Omega^{\rho_1} \divideontimes \Omega^{\rho_2} = \Omega^{\rho_1 \times \rho_2}$.

Example 3.2 *The function $G = (\lambda x^{\mathsf{nat}}.\lambda y^{\mathsf{nat} \times \mathsf{nat}}.\mathsf{prj}_1 y)z$ has, under the assumption Ω^{nat} for the free variable z, the property $(\Delta^{\mathsf{nat}} \divideontimes \Omega^{\mathsf{nat}}) \twoheadrightarrow (\Delta^{\mathsf{nat}} \divideontimes \Delta^{\mathsf{nat}})$. In fact, z is not used and the second component of any pair of numbers supplied to G is not used.*

The set-theoretic intersection between p.e.r.'s over a type ρ is still a p.e.r. over ρ and it represents the logical conjunction between properties, i.e., if a program P has both the properties $\mathcal{R}_1$ and $\mathcal{R}_2$, then P has also the property $\mathcal{R}_1 \cap \mathcal{R}_2$ (and vice-versa). Moreover the set-theoretic inclusion between p.e.r.'s over a type ρ represents a logical implication between properties, i.e., if $\mathcal{R}_1 \subseteq \mathcal{R}_2$ and a program P has the property $\mathcal{R}_1$, then P has also the property $\mathcal{R}_2$. For instance, for every p.e.r. $\mathcal{R}$ over ρ, $\mathcal{R} \subseteq \Omega^\rho$, according to the fact that Ω^ρ represents the "true" property. We also have that $\Omega^{\rho_1} \twoheadrightarrow \Delta^{\rho_2} \subseteq \Delta^{\rho_1} \twoheadrightarrow \Delta^{\rho_2}$, according to fact that if the application of a function can be used without using the argument, then it can still be used if the value of the argument can be used.

4 Evaluation types

In this section we introduce the set of conjunctive non-standard types for useless-code analysis, that we call *evaluation types* (*e-types* for short).

4.1 E-types syntax and semantics

The syntax of the e-types does not allow conjunctions to the right of the arrow type constructor. This restriction, which is essentially the one proposed in [18], makes the type manipulation easier without affecting the power of the logic.

Definition 4.1 (Evaluation types) *The set of* e-types $\mathbf{L}$*, ranged over by* ϕ *and* ψ*, is defined by:* $\mathbf{L} = \cup_{\rho \in \mathbf{T}} \mathbf{L}(\rho)$*, where* $\mathbf{L}(\rho)$ *is the sets of the* e-types with underlying type ρ*, defined by the rules in Fig. 3.* Conjunctions *of elements in* $\mathbf{L}(\rho)$*, i.e., expressions of the shape* $\phi_1 \wedge \cdots \wedge \phi_n$*, for some* $n \geq 1$ *and* $\phi_1, \ldots, \phi_n \in \mathbf{L}(\rho)$*, are ranged over by* ξ *and* ν*.*

In the following we will often write, since no confusion may arise, δ and ω as shorts for δ^{nat} and ω^{nat}, respectively. Let ϕ^ρ (ξ^ρ) range over e-types (conjunctions) with underlying type ρ.

$$(\delta)\ \delta^{\mathsf{nat}} \in \mathbf{L}(\mathsf{nat}) \qquad (\omega)\ \omega^\rho \in \mathbf{L}(\rho) \qquad (\rightarrow)\ \frac{\phi_1, \ldots, \phi_n \in \mathbf{L}(\rho) \quad \psi \in \mathbf{L}(\sigma)}{\phi_1 \wedge \cdots \wedge \phi_n \rightarrow \psi \in \mathbf{L}(\rho \rightarrow \sigma)} \begin{cases} n \geq 1 \\ \text{and } \psi \neq \omega^\sigma \end{cases}$$

$$(\times)\ \frac{\psi_1 \in \mathbf{L}(\rho_1) \quad \psi_2 \in \mathbf{L}(\rho_2)}{\psi_1 \times \psi_2 \in \mathbf{L}(\rho_1 \times \rho_2)}\ \exists i \in \{1,2\}.\psi_i \neq \omega^{\rho_i}$$

Figure 3: Evaluation types

The meaning of an e-type ϕ^ρ is given by interpreting it as a p.e.r., $[\![\phi^\rho]\!]$, over the type ρ (see Section 3).

Definition 4.2 (Semantics of e-types) *The interpretation* $[\![\phi]\!]$ *of an e-type* ϕ *is defined by:*

$$\begin{array}{rcl@{\qquad}rcl} [\![\delta^{\mathsf{nat}}]\!] & = & \Delta^{\mathsf{nat}} & [\![\omega^\rho]\!] & = & \Omega^\rho \\ [\![\xi \rightarrow \psi]\!] & = & [\![\xi]\!] \twoheadrightarrow [\![\psi]\!] & [\![\psi_1 \times \psi_2]\!] & = & [\![\psi_1]\!] \divideontimes [\![\psi_2]\!] \\ [\![\phi_1 \wedge \cdots \wedge \phi_n]\!] & = & \bigcap_{1 \leq i \leq n} [\![\phi_i]\!] & & & \end{array}$$

The e-type syntax (Definition 4.1) has been designed in such a way that syntactically different *conjunction-free* e-types denote (Definition 4.2) different p.e.r.'s. This observation justifies the choice of not having e-types of the form δ^ρ, $\xi \rightarrow \omega^\sigma$ and $\omega^{\sigma_1} \times \omega^{\sigma_2}$, where $\rho \neq \mathsf{nat}$, $\sigma, \sigma_1, \sigma_2$ are any types, and ξ is any conjunction.

Definition 4.3 (ω-e-types and δ-e-types)

1. *We call* ω*-e-types the e-types in the set* $\mathbf{L}_\omega = \{\omega^\rho \mid \rho \in \mathbf{T}\}$*.*
2. *The set of the* δ*-e-types (*$\mathbf{L}_\delta$*) is the subset of the e-types which do not contain subexpressions of the form* ω^ρ *(for some* ρ*). For every type* $\rho \in \mathbf{T}$*, let* $\delta(\rho)$ *be the corresponding unique conjunction-free* δ*-e-type.*

The following proposition shows that ω-e-types and δ-e-types provide a syntactical characterization for the set of the e-types ϕ^ρ such that $[\![\phi^\rho]\!] = \Omega^\rho$ and for the set of the e-types ψ^σ such that $[\![\psi^\sigma]\!] = \Delta^\sigma$, respectively.

Proposition 4.4 *1. $\phi^\rho \in \mathbf{L}_\omega$ (i.e. $\phi = \omega^\rho$) if and only if $[\![\phi^\rho]\!] = \Omega^\rho$, and*

2. $\phi^\rho \in \mathbf{L}_\delta$ implies $[\![\phi^\rho]\!] = \Delta^\rho$.

4.2 An entailment relation for e-types

In this section introduce an entailment relation between e-types, $\leq$, which models the set-theoretic inclusion between the interpretation of the e-types.

Definition 4.5 (Entailment relation $\leq$) *We write $\xi_1 \leq \xi_2$ to mean that this judgement is derivable by the rules in Fig. 4. By $\cong$ we denote the equivalence relation induced by $\leq$.*

$$(\delta)\ \delta^{\mathsf{nat}} \leq \delta^{\mathsf{nat}} \qquad (\omega)\ \phi^\rho \leq \omega^\rho \qquad (\to)\ \frac{\xi_2 \leq \xi_1 \quad \psi_1 \leq \psi_2}{\xi_1 \to \psi_1 \leq \xi_2 \to \psi_2}\ \psi_2 \not\cong \mathbf{L}_\omega$$

$$(\times)\ \frac{\phi_1 \wedge \cdots \wedge \phi_n \leq \psi_1 \wedge \cdots \wedge \psi_n \quad \phi'_1 \wedge \cdots \wedge \phi'_n \leq \psi'_1 \wedge \cdots \wedge \psi'_n}{\phi_1 \times \phi'_1 \wedge \cdots \phi_n \times \phi'_n \leq \psi_1 \times \psi'_1 \wedge \cdots \psi_n \times \psi'_n} \begin{cases} \forall i \in \{1,\ldots,n\}. \\ \{\phi_i, \phi'_i\} \not\subseteq \mathbf{L}_\omega \\ \text{and } \{\psi_i, \psi'_i\} \not\subseteq \mathbf{L}_\omega \end{cases}$$

$$(\wedge)\ \frac{{\phi_{j_1}}^\rho \leq {\psi_1}^\rho \cdots {\phi_{j_n}}^\rho \leq {\psi_m}^\rho}{{\phi_1}^\rho \wedge \cdots \wedge {\phi_n}^\rho \leq {\psi_1}^\rho \wedge \cdots \wedge {\psi_m}^\rho}\ j_1,\ldots,j_m \in \{1,\ldots,n\}$$

Figure 4: Entailment rules for e-types and conjunctions (system $\leq$)

Note that $\leq$ is reflexive and transitive. The most interesting rules are $(\wedge)$ and $(\times)$. Rule $(\wedge)$ subsumes the standard rules of conjunction introduction and elimination. Rule $(\times)$ allows to compare conjunctions of $n \geq 1$ products. Note that, restricting the rule to e-types (i.e., requiring $n = 1$), reduces the power of the system. For instance, we have both $\delta \wedge \omega \leq \delta \wedge \delta$ and $\omega \wedge \delta \leq \delta \wedge \delta$, so by rule $(\times)$, we get $\delta \times \omega \wedge \omega \times \delta \leq \delta \times \delta \wedge \delta \times \delta$ (and so: $\delta \times \omega \wedge \omega \times \delta \cong \delta \times \delta$), which is valid (since $[\![(\phi_1 \times \psi_1) \wedge \cdots\cdots \wedge (\phi_n \times \psi_n)]\!] = [\![\phi_1 \wedge \cdots \wedge \phi_n]\!] \bbbar [\![\psi_1 \wedge \cdots \wedge \psi_n]\!]$) but can not be derived when rule $(\times)$ is restricted to e-types.

Theorem 4.6 (Soundness of $\leq$) *$\phi \leq \psi$ implies $[\![\phi]\!] \subseteq [\![\psi]\!]$.*

The entailment rules are not complete. Take, for instance, $\phi_1 = \omega \to \delta \to \delta$, $\phi_2 = \delta \to \omega \to \delta$, and $\psi = \omega \to \omega \to \delta$. We have $[\![\phi_1 \wedge \phi_2]\!] = [\![\psi]\!]$, but $\phi_1 \wedge \phi_2 \not\leq \psi$.

5 Using e-types for useless-code elimination

The basic idea for using e-types to detect the useless-code in a given PCFP term M is the following: if we can prove that (under suitable assumptions on its free variables) the term has a δ-e-type while some of its subterms have ω-e-types, then the value of M can be computed without using these subterms, so they are useless-code.

5.1 An e-type assignment system for detecting useless-code

An assumption for a term variable x of type ρ is an expression of the shape $x : \xi$, where ξ is a conjunction of underlying type ρ. A basis is a set Σ of assumptions $x : \xi$ for term variables such that each term variable occurs at most once in Σ. E-types are assigned to terms $M \in \Lambda_{\mathbf{T}}$ by a set of inference rules for judgements of the form $\Sigma \vdash M'$ where M' is a *decoration* of the term M, i.e., it has written in it (some of) the e-types assigned to its subterms, and Σ is a basis containing an assumption for each free variable of the term M.

For any e-type ϕ (conjunction ξ) let $\epsilon(\phi)$ ($\epsilon(\xi)$) denote the underlying type of ϕ (ξ), and for any basis Σ let $\epsilon(\Sigma)$ denote the set of terms variables occurring in Σ, i.e., $\epsilon(\Sigma) = \{x \mid x : \xi \in \Sigma, \text{ for some } \xi\}$. Let $\Sigma, x : \xi$ denote the basis $\Sigma \cup \{x : \xi\}$ where it is assumed that x does not appear in Σ.

Definition 5.1 (E-type assignment system $\vdash$) *We write $\Sigma \vdash M^{\phi}$ to mean that this judgement can be derived by the rules in Fig. 5.*

A sequence $\{N^{\phi_1}; \cdots; N^{\phi_n}\}$, is just a way of keeping track of $n \geq 1$ decorations of a subterm. These decorations correspond to different uses of the subterm. When $n = 1$ we will often write N^{ϕ_1} instead of $\{N^{\phi_1}\}$. Let $\Lambda^{\vdash}_{\mathbf{T}}$ be the set of $\vdash$-decorations,

$$(\omega)\ \frac{\epsilon(\Sigma) \supseteq \mathrm{FV}(M) \quad M \in \Lambda_{\rho}}{\Sigma \vdash M^{\omega^{\rho}}} \qquad (\mathrm{Con})\ \Sigma \vdash \mathsf{k}^{\delta(\mathbf{T}(\mathsf{k}))}$$

$$(\mathrm{Var})\ \frac{\phi_i \leq \phi}{\Sigma, x : \phi_1 \wedge \cdots \wedge \phi_n \vdash x^{\phi}}\ i \in \{1, \ldots, n\} \text{ and } \phi \notin \mathbf{L}_{\omega}$$

$$(\to \mathrm{I})\ \frac{\Sigma, x : \phi_1 \wedge \cdots \wedge \phi_n \vdash M^{\psi}}{\Sigma \vdash (\lambda x.M)^{\phi_1 \wedge \cdots \wedge \phi_n \to \psi}}\ \psi \notin \mathbf{L}_{\omega}$$

$$(\to \mathrm{E})\ \frac{\Sigma \vdash M^{\phi_1 \wedge \cdots \wedge \phi_n \to \psi} \quad \Sigma \vdash N^{\phi_1} \quad \cdots \quad \Sigma \vdash N^{\phi_n}}{\Sigma \vdash (M\{N^{\phi_1}; \cdots; N^{\phi_n}\})^{\psi}}$$

$$(\times \mathrm{I})\ \frac{\Sigma \vdash {M_1}^{\psi_1} \quad \Sigma \vdash {M_2}^{\psi_2}}{\Sigma \vdash \langle M_1, M_2 \rangle^{\psi_1 \times \psi_2}}\ \exists i \in \{1, 2\}.\psi_i \notin \mathbf{L}_{\omega}$$

$$(\times \mathrm{E}_i)\ \frac{\Sigma \vdash M^{\psi_1 \times \psi_2}}{\Sigma \vdash (\mathsf{prj}_i M^{\psi_{3-i}})^{\psi_i}}\ i \in \{1, 2\} \text{ and } \psi_i \notin \mathbf{L}_{\omega}$$

$$(\mathrm{Fix})\ \frac{\Sigma, x : \phi_1 \wedge \cdots \wedge \phi_n \vdash M^{\phi_1} \quad \cdots \quad \Sigma, x : \phi_1 \wedge \cdots \wedge \phi_n \vdash M^{\phi_n}}{\Sigma \vdash (\mathsf{fix}\, x.\{M^{\phi_1}; \cdots; M^{\phi_n}\})^{\phi_1}}\ \forall i \in \{1, \ldots, n\}.\phi_i \notin \mathbf{L}_{\omega}$$

$$(\mathrm{If})\ \frac{\Sigma \vdash N^{\delta^{\mathsf{nat}}} \quad \Sigma \vdash {M_1}^{\phi_1} \quad \Sigma \vdash {M_2}^{\phi_2} \quad \phi_1 \leq \psi \quad \phi_2 \leq \psi}{\Sigma \vdash (\mathsf{ifz}\, N\, \mathsf{then}\, {M_1}^{\phi_1}\, \mathsf{else}\, {M_2}^{\phi_2})^{\psi}}\ \psi \notin \mathbf{L}_{\omega}$$

Figure 5: Rules for e-type assignment (system $\vdash$)

i.e., $\Lambda^{\vdash}_{\mathbf{T}} = \{M' \mid \Sigma \vdash M' \text{ for some basis } \Sigma\}$. For any decoration $M' \in \Lambda^{\vdash}_{\mathbf{T}}$, let $\epsilon(M')$ denote the underlying PCFP term and, for every term $M \in \Lambda_{\mathbf{T}}$, let $\Lambda^{\vdash}_{\mathbf{T}}(M) = \{M' \in \Lambda^{\vdash}_{\mathbf{T}} \mid \epsilon(M') = M\}$.

For any finite set Γ of term variables let $\delta(\Gamma)$ and $\omega(\Gamma)$ denote the bases $\{x^\sigma : \delta(\sigma) \mid x^\sigma \in \Gamma\}$ and $\{x^\sigma : \omega^\sigma \mid x^\sigma \in \Gamma\}$, respectively. For any term $M \in \Lambda_{\mathbf{T}}$, define $\delta(M)$ as the decoration of M in which each subterm is annotated by the corresponding conjunction-free δ-e-type. It is immediate to verify that $M \in \Lambda_\rho$ implies both $\delta(\mathrm{FV}(M)) \vdash \delta(M)$ and, for all basis Σ such that $\epsilon(\Sigma) \supseteq \mathrm{FV}(M)$, $\Sigma \vdash M^{\omega^\rho}$.

To state the soundness of the e-type assignment system we introduce the following definition.

Definition 5.2 *1. Two environments e_1, e_2 are Σ-related if and only if, for all $x : \xi \in \Sigma$, $\langle e_1(x), e_2(x)\rangle \in [\![\xi]\!]$.*

2. Let $\Sigma \vdash M^\psi$ and $\Sigma \vdash N^\psi$. We write $\epsilon(M^\psi) \sim^{\Sigma}_{\psi} \epsilon(N^\psi)$ to mean that for all e_1, e_2, if e_1 and e_2 are Σ-related, then $\langle [\![\epsilon(M^\psi)]\!]_{e_1}, [\![\epsilon(N^\psi)]\!]_{e_2}\rangle \in [\![\psi]\!]$.

Now we can state the soundness of the system $\vdash$.

Theorem 5.3 (Soundness of $\vdash$) *Let $\Sigma \vdash P^\chi$. Then $\epsilon(P^\chi) \sim^{\Sigma}_{\chi} \epsilon(P^\chi)$.*

Roughly speaking, this theorem means that, given a term M, if we have a derivation of $\Sigma \vdash M'^\chi$ in which the subderivation associated to a subterm P of M is $\Sigma' \vdash P'^\phi$, then if we replace P by any term $\sim^{\Sigma'}_{\phi}$-related to it we get a term $\sim^{\Sigma}_{\chi}$-related to M.

We give a first example of useless-code that can be detected by system $\vdash$, showing how conjunction is used, in rules ($\to$ I) and ($\to$ E), to assign different e-types to the argument of a function in order to exploit its different uses in the function's body.

Example 5.4 (Use of rules ($\to$ I) and ($\to$ E)) *Take the term of type* nat,

$$M \;=\; (\lambda f^{(\mathsf{nat}\to\mathsf{nat})\to\mathsf{nat}\to\mathsf{nat}}.+\langle f(\lambda x^{\mathsf{nat}}.3)P, f(\lambda y^{\mathsf{nat}}.y)Q\rangle)\;\; (\lambda h^{\mathsf{nat}\to\mathsf{nat}}.\lambda z^{\mathsf{nat}}.hz)\;\;,$$

where $\mathrm{FV}(M) = \mathrm{FV}(P) = \{z^{\mathsf{nat}}\}$ *and* $\mathrm{FV}(Q) = \emptyset$.

Let $\phi_1 = (\omega \to \delta) \to \omega \to \delta$ and $\phi_2 = (\delta \to \delta) \to \delta \to \delta$. The $\vdash$-typing $\Sigma \vdash M'$, where $\Sigma = \{z : \omega\}$ and M' is the decoration

$$((\lambda f.+\langle f(\lambda x.3)^{\omega\to\delta}P^\omega, f(\lambda y.y)^{\delta\to\delta}Q^\delta\rangle^{\delta\times\delta})\;\; \{(\lambda h.\lambda z.hz^\omega)^{\phi_1}; (\lambda h.\lambda z.hz^\delta)^{\phi_2}\})^\delta\;\;,$$

shows that the value of M can be computed without using the subterm P. In fact, we can replace P (which has e-type ω) by any other term of type nat *without changing the meaning of M (which has e-type δ).*

Note that, since it does not exist a conjunction-free *e-type ϕ which entails both ϕ_1 and ϕ_2, the useless-code P cannot be detected without using conjunction.*

In order to use the system $\vdash$ to detect useless-code we identify a subset of $\vdash$-typings (that we call *faithful*) for which the $\sim^{\Sigma}_{\phi}$ relations implies the $\simeq_{\mathrm{obs}}$ relation. A faithful $\vdash$-typing says that the term can be used (it has a δ-e-type) and that any its free variable is either used (it has a δ-e-type) or not used at all (it has an ω-e-type).

Definition 5.5 (Faithful $\vdash$-typing) *A $\vdash$-typing of a term M of type ρ, $\Sigma \vdash M'^\phi$, is a* faithful $\vdash$*-typing and M'^ϕ is a faithful $\vdash$-decoration, if $\epsilon(\Sigma) = \mathrm{FV}(M)$, $\phi = \delta(\rho)$, and, for all assumptions $x^\sigma : \nu \in \Sigma$, either $\nu = \delta(\sigma)$ or $\nu = \omega^\sigma$.*

The fundamental property of faithful $\vdash$-typings is the following.

Lemma 5.6 *Let $\Sigma_1 \vdash {M'}^{\delta(\rho)}$ and $\Sigma_2 \vdash {N'}^{\delta(\rho)}$ be faithful $\vdash$-typings (of terms M and N, respectively) such that Σ_1 and Σ_2 contains the same assumptions for the variables in $\mathrm{FV}(M) \cap \mathrm{FV}(N)$. Then $M \sim_{\phi}^{\Sigma_1 \cup \Sigma_2} N$ implies $M \simeq_{\mathrm{obs}} N$.*

Example 5.7 *The typing $\Sigma \vdash M'$ in Example 5.4 is faithful. So, according to Lemma 5.6, M is observationally equivalent to every term obtained by replacing the subterm P (which has e-type ω) by any term of type* nat.

5.2 A mapping for removing useless-code

In this section we define a useless-code elimination mapping, **O**, that takes a faithful $\vdash$-decoration M^{ψ} and simplifies it by replacing the useless-code showed by the decorations with "dummy variables".

5.2.1 Dummy variables and pruning relation

For every type ρ, we consider a countable set $\{ \mathsf{d}^{\rho}, \mathsf{d}_1^{\rho}, \mathsf{d}_2^{\rho}, \ldots, \}$ of *dummy variables* of type ρ. We remark that dummy variables are not present in the original programs: they are introduced by the useless-code elimination mapping **O** as place-holders for the useless-code removed. So, it is safe to assume that every dummy variable is useless-code and that all the occurrences of dummy variables in a program are free and distinct.

For every term M, let $\mathrm{DV}(M)$ be the set of the dummy variables in M, and let $\mathrm{FNDV}(M) = \mathrm{FV}(M) - \mathrm{DV}(M)$.

The pruning relation on terms allows to compare simplifications of a given term M.

Definition 5.8 (Pruning relation) *Let $M \in \Lambda_{\rho}$. We say that a term N is a* pruning *of M, and write $M \preceq_{\mathrm{prn}} N$, if N can be obtained from M by replacing some subterms by fresh dummy variables of the corresponding type.*

It is immediate to verify that, for every term M, if $M \in \Lambda_{\rho}$ and $M \preceq_{\mathrm{prn}} N$ then $N \in \Lambda_{\rho}$ and $\mathrm{FNDV}(N) \subseteq \mathrm{FNDV}(M)$. Let $\Lambda_{\mathbf{T}}^{\preceq_{\mathrm{prn}}}(M)$ be the set of the prunings of M, i.e., $\Lambda_{\mathbf{T}}^{\preceq_{\mathrm{prn}}}(M) = \{N \mid M \preceq_{\mathrm{prn}} N\}$. In the following we will identify the prunings of a term M modulo renaming of the dummy variables. So, modulo this identification, we have that for every term M the set $\Lambda_{\mathbf{T}}^{\preceq_{\mathrm{prn}}}(M)$ is finite.

Example 5.9 *Take the term $P = (\lambda x^{\mathsf{nat}}.3)5$. The set $\Lambda_{\mathbf{T}}^{\preceq_{\mathrm{prn}}}(P)$ has (modulo renaming of the dummy variables) 7 elements and forms a complete lattice (showed in Fig. 6).*

Definition 5.10 (Operation inf) *For every term M of type ρ the binary operation* **inf** *on $\Lambda_{\mathbf{T}}^{\preceq_{\mathrm{prn}}}(M)$ is inductively defined by the clause*

$$\mathbf{inf}(M, \mathsf{d}^{\rho}) = \mathbf{inf}(\mathsf{d}^{\rho}, M) = M, \text{ for every term } M \text{ of type } \rho$$

together with standard clauses specifying that **inf** *commutes with the term-formation operators.*

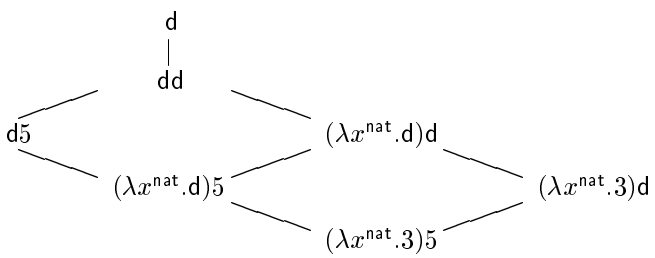

Figure 6: The lattice $\Lambda_{\mathbf{T}}^{\preceq_{\mathrm{prn}}}(P)$

.

It is immediate to verify that the operation **inf** is idempotent, commutative, and associative. Moreover the following property holds.

Theorem 5.11 *For every term M of type ρ, the set $\Lambda_{\mathbf{T}}^{\preceq_{\mathrm{prn}}}(M)$ is a finite complete lattice with order relation $\preceq_{\mathrm{prn}}$, meet operation* **inf**, *bottom element M, and top element d^{ρ}.*

5.2.2 The simplification mapping

Definition 5.12 (Simplification mapping O) *The function $\mathbf{O} : \Lambda_{\mathbf{T}}^{\vdash} \to \Lambda_{\mathbf{T}}$ is defined by the clauses in Fig. 7, where the occurrence of "d" (first row) denotes a fresh dummy variable of the proper type.*

The simplification mapping produces well typed terms and, for every $\vdash$-typing of M, $\Sigma \vdash M'$, we have that $M \preceq_{\mathrm{prn}} \mathbf{O}(M')$. Moreover the simplification mapping is correct w.r.t. the e-type semantics, as the following property shows.

Lemma 5.13 (Correctness of O) *For every term M, if $\Sigma \vdash M'^{\psi}$ is a $\vdash$-typing of M and N is term such that $M \preceq_{\mathrm{prn}} N \preceq_{\mathrm{prn}} \mathbf{O}(M'^{\psi})$, then*

1. *$\Sigma' \vdash N'^{\psi}$, where $\Sigma' = \{x : \xi \in \Sigma \mid x \in \mathrm{FNDV}(N)\} \cup \{\mathsf{d}^{\sigma} : \omega^{\sigma} \mid \mathsf{d}^{\sigma} \in \mathrm{DV}(N)\}$ and N'^{ψ} is the decoration of N obtained from M'^{ψ} by replacing by d the subterms which have been replaced by d in N.*
2. *$M \sim_{\psi}^{\Sigma''} N$, where $\Sigma'' = \Sigma \cup \{\mathsf{d}^{\sigma} : \omega^{\sigma} \mid \mathsf{d}^{\sigma} \in \mathrm{DV}(N)\}$.*

The system $\vdash$ and the simplification mapping $\mathbf{O}$ can be used to simplify terms into observationally equivalent ones: in fact the dummy variables introduced by applying the mapping $\mathbf{O}$ on a faithful $\vdash$-typing of M are useless-code.

Theorem 5.14 (O on faithful $\vdash$-typings preserves $\simeq_{\mathrm{obs}}$) *Let $\Sigma \vdash M^{\delta(\rho)}$ be a faithful $\vdash$-typing. Then $\epsilon(M^{\delta(\rho)}) \simeq_{\mathrm{obs}} \mathbf{O}(M^{\delta(\rho)})$.*

Example 5.15 *The $\vdash$-typing $\Sigma \vdash M'$ of Examples 5.4 and 5.7 is faithful. By applying the simplification mapping $\mathbf{O}$ to M' we get the term*

$$N \;=\; (\lambda f^{(\mathsf{nat}\to\mathsf{nat})\to\mathsf{nat}\to\mathsf{nat}}.+\langle f(\lambda x^{\mathsf{nat}}.3)\mathsf{d}, f(\lambda y^{\mathsf{nat}}.y)Q\rangle)\;\;(\lambda h^{\mathsf{nat}\to\mathsf{nat}}.\lambda z^{\mathsf{nat}}.hz)\;\;,$$

which is observationally equivalent to M.

$\mathbf{O}(M^{\psi}) = \mathsf{d}, \quad \text{if } \psi \in \mathbf{L}_{\omega}, \quad \text{otherwise:}$

$\mathbf{O}(\mathsf{k}^{\psi}) = \mathsf{k}$

$\mathbf{O}(x^{\psi}) = x$

$\mathbf{O}(\langle M_1, M_2\rangle^{\psi_1\times\psi_2}) = \langle \mathbf{O}({M_1}^{\psi_1}), \mathbf{O}({M_2}^{\psi_2})\rangle$

$\mathbf{O}((\mathsf{prj}_i M^{\psi_{3-i}})^{\psi_i}) = \mathsf{prj}_i \mathbf{O}(M^{\psi_1\times\psi_2}), \quad \text{where } i \in \{1,2\}$

$\mathbf{O}((M\{N^{\phi_1};\cdots;N^{\phi_n}\})^{\psi}) = \mathbf{O}(M^{\phi_1\wedge\cdots\wedge\phi_n\to\psi})\mathbf{inf}(\mathbf{O}(N^{\phi_1}),\ldots,\mathbf{O}(N^{\phi_n}))$

$\mathbf{O}((\lambda x.M)^{\phi_1\wedge\cdots\wedge\phi_n\to\psi}) = \lambda x^{\epsilon(\phi_1)}.\mathbf{O}(M^{\psi})$

$\mathbf{O}((\mathsf{fix}\, x.\{M^{\phi_1};\cdots;M^{\phi_n}\})^{\phi_1}) = \mathsf{fix}\, x.\mathbf{inf}(\mathbf{O}(M^{\phi_1}),\ldots,\mathbf{O}(M^{\phi_n}))$

$\mathbf{O}((\mathsf{ifz}\, N\, \mathsf{then}\, M_1^{\phi_1}\, \mathsf{else}\, M_2^{\phi_2})^{\psi}) = \mathsf{ifz}\, \mathbf{O}(N^{\delta^{\mathsf{nat}}})\, \mathsf{then}\, \mathbf{O}({M_1}^{\phi_1})\, \mathsf{else}\, \mathbf{O}({M_2}^{\phi_2})$

Figure 7: The simplification mapping $\mathbf{O}$

The next example illustrates the other essential use of conjunction, that is in rule (Fix), to analyze recursive terms .

Example 5.16 (Use of rule (Fix)**)** *Take the type* $\rho = (\mathsf{nat}\to\mathsf{nat})\to\mathsf{nat}\to\mathsf{nat}\to\mathsf{nat}\to\mathsf{nat}$ *and the recursive term* $\mathsf{fix}\, f^{\rho}.M$*, where*

$$M = \lambda g^{\,\mathsf{nat}\to\mathsf{nat}}.\lambda x^{\,\mathsf{nat}}.\lambda y^{\,\mathsf{nat}}.\lambda z^{\,\mathsf{nat}}.\mathsf{ifz}\ z\ \mathsf{then}\ gx\ \mathsf{else}\ f(\lambda u^{\mathsf{nat}}.3)yx(-\langle z,1\rangle) \quad .$$

Let $\phi_1 = (\delta\to\delta)\to\delta\to\omega\to\delta\to\delta$ *and* $\phi_2 = (\omega\to\delta)\to\omega\to\omega\to\delta\to\delta$*. We have that* $\emptyset \vdash (\mathsf{fix}\, f.\{M^{\phi_1}; M^{\phi_2}\})^{\phi_1}$*, where*

$$\begin{array}{rcl} M^{\phi_1} & = & (\lambda g.\lambda x.\lambda y.\lambda z.\mathsf{ifz}\ z^{\delta}\ \mathsf{then}\ gx^{\delta}\ \mathsf{else}\ f(\lambda u.3)^{\omega\to\delta}y^{\omega}x^{\omega}(-\langle z,1\rangle^{\delta\times\delta})^{\delta})^{\phi_1} \\ M^{\phi_2} & = & (\lambda g.\lambda x.\lambda y.\lambda z.\mathsf{ifz}\ z^{\delta}\ \mathsf{then}\ gx^{\omega}\ \mathsf{else}\ f(\lambda u.3)^{\omega\to\delta}y^{\omega}x^{\omega}(-\langle z,1\rangle^{\delta\times\delta})^{\delta})^{\phi_2} \quad . \end{array}$$

So, when considering the application of $\mathsf{fix}\, f^{\rho}.M$ *to any closed terms* Q_1*,* Q_2*,* Q_3 *of type* nat*:* $R = (\mathsf{fix}\, f^{\rho}.M)(\lambda v^{\mathsf{nat}}.v)Q_1Q_2Q_3$*, we have the faithful* $\vdash$*-typing:* $\emptyset \vdash (\mathsf{fix}\, f.\{M^{\phi_1}; M^{\phi_2}\})(\lambda v.v)^{\delta\to\delta}Q_1^{\delta}Q_2^{\omega}Q_3^{\delta})^{\delta}$*, and, by applying the simplification mapping* $\mathbf{O}$*, get the observationally equivalent term:* $S = (\mathsf{fix}\, f^{\rho}.N)(\lambda v^{\mathsf{nat}}.v)Q_1\mathsf{d}_3^{\mathsf{nat}}Q_3$*, where*

$$N = \lambda g^{\,\mathsf{nat}\to\mathsf{nat}}.\lambda x^{\,\mathsf{nat}}.\lambda y^{\,\mathsf{nat}}.\lambda z^{\,\mathsf{nat}}.\mathsf{ifz}\ z\ \mathsf{then}\ gx\ \mathsf{else}\ f(\lambda u^{\mathsf{nat}}.3)\mathsf{d}_1^{\mathsf{nat}}\mathsf{d}_2^{\mathsf{nat}}(-\langle z,1\rangle) \quad .$$

Note that there is a (conjunction-free) derivation of $\emptyset \vdash (\mathsf{fix}\, f.M^{\phi_2})^{\phi_2}$*, but there is* no *derivation for* $\emptyset \vdash (\mathsf{fix}\, f.M^{\phi_1})^{\phi_1}$*. Indeed the above simplification of* R *cannot be done without using conjunction.*

5.3 Optimal faithful $\vdash$-typings and optimal simplifications

In this section we prove that for every term M there is an *optimal* faithful $\vdash$-typing, i.e., a typing which allows the mapping $\mathbf{O}$ to remove from M all the useless-code that can be proved by the system $\vdash$.

5.3.1 Optimal faithful $\vdash$-typings

Given two faithful $\vdash$-typings of M, $\Sigma_1 \vdash M_1^{\delta(\rho)}$ and $\Sigma_2 \vdash M_2^{\delta(\rho)}$, the one which proves more useless-code is the one that assigns ω-e-types to more subterms. This fact induces a preorder relation (that we will denote by $\sqsubseteq$) on the set of the $\vdash$-typings of M. The preorder $\sqsubseteq$ (for e-types, basis, $\vdash$-decorations, and $\vdash$-typings) is defined as follows.

Definition 5.17 (The $\sqsubseteq$ preorder) *For every type $\rho \in \mathbf{T}$ the preorder $\sqsubseteq$ on the set of e-types $\mathbf{L}(\rho)$ is defined by the rules in Fig. 8* [2].

For every set of term variables Γ the preorder $\sqsubseteq$ on the set of basis $\{\Sigma \mid \epsilon(\Sigma) = \Gamma\}$ is defined by: $\Sigma_1 \sqsubseteq \Sigma_2$ if, for every $x^\rho : \nu_2 \in \Sigma_2$, there exists $x^\rho : \nu_1 \in \Sigma_1$ such that $\nu_1 \sqsubseteq \nu_2$.

For every term M the preorder $\sqsubseteq$ on the set of $\vdash$-decorations $\Lambda^{\vdash}_{\mathbf{T}}(M)$ is defined by the rules in Fig. 9.

For every term M and for every set of term variables Γ such that $\Gamma \supseteq \mathrm{FV}(M)$, the preorder $\sqsubseteq$ on the set of the $\vdash$-typings of M in which the basis contains exactly the variables in Γ is defined by: $\Sigma_1 \vdash M_1^{\phi_1} \sqsubseteq \Sigma_2 \vdash M_2^{\phi_2}$ if $\Sigma_1 \sqsubseteq \Sigma_2$ and $M_1^{\phi_1} \sqsubseteq M_2^{\phi_2}$.

Let $\equiv$ be the equivalence relation induced by $\sqsubseteq$.

$$(\omega)\ \phi^\rho \sqsubseteq \omega^\rho \qquad (\delta)\ \delta^{\mathsf{nat}} \sqsubseteq \delta^{\mathsf{nat}} \qquad (\to)\ \frac{\xi_1 \sqsubseteq \xi_2 \quad \psi_1 \sqsubseteq \psi_2}{\xi_1 \to \psi_1 \sqsubseteq \xi_2 \to \psi_2}\ \phi_2, \psi_2 \notin \mathbf{L}_\omega$$

$$(\times)\ \frac{\phi_1 \sqsubseteq \psi_1 \quad \phi_2 \sqsubseteq \psi_2}{\phi_1 \times \phi_2 \sqsubseteq \psi_1 \times \psi_2}\ \exists i \in \{1,2\}.\phi_i, \psi_i \notin \mathbf{L}_\omega$$

$$(\wedge)\ \frac{\phi_{j_1} \sqsubseteq \psi_1 \ \cdots\ \phi_{j_m} \sqsubseteq \psi_m}{\phi_1 \wedge \cdots \wedge \phi_n \sqsubseteq \psi_1 \wedge \cdots \wedge \psi_m}\ j_1, \ldots, j_m \in \{1, \ldots, n\}$$

Figure 8: Relation $\sqsubseteq$ on $\mathbf{L}(\rho)$

The following property says that "bigger" faithful $\vdash$-typing prove more useless-code.

Lemma 5.18 *For every term P, if $\Sigma_1 \vdash P_1{}^{\chi_1}$ and $\Sigma_2 \vdash P_2{}^{\chi_2}$ two $\vdash$-typings of P such that $\Sigma_1 \vdash P_1{}^{\chi_1} \sqsubseteq \Sigma_2 \vdash P_2{}^{\chi_2}$, then $\mathbf{O}(P_1{}^{\chi_1}) \preceq_{\mathrm{prn}} \mathbf{O}(P_2{}^{\chi_2})$.*

For every $M \in \Lambda_\rho$, consider the set of equivalence classes of the relation $\equiv$:

$$\mathbf{faithful}^{\vdash}(M)/\equiv \ = \ \{[\Sigma \vdash M'^{\delta(\rho)}]^{\equiv} \mid \Sigma \vdash M'^{\delta(\rho)} \text{ is a faithful } \vdash\text{-typing of } M\}\ .$$

[2] The preorder $\sqsubseteq$, which is covariant in both the arguments of the constructor $\to$, should not be confused with the e-type entailment relation $\leq$, which is contravariant in the left argument of $\to$.

$$(\omega)\ M^{\phi} \sqsubseteq \epsilon(M)^{\omega^{\epsilon(\phi)}} \qquad (\mathrm{Var})\ \frac{\phi \sqsubseteq \psi}{x^{\phi} \sqsubseteq x^{\psi}}\ \psi \notin \mathbf{L}_{\omega} \qquad (\mathrm{Con})\ \mathsf{k}^{\delta(\mathbf{T}(\mathsf{k}))} \sqsubseteq \mathsf{k}^{\delta(\mathbf{T}(\mathsf{k}))}$$

$$(\to \mathrm{I})\ \frac{\xi \sqsubseteq \nu \qquad M^{\phi} \sqsubseteq N^{\psi}}{(\lambda x.M)^{\xi \to \phi} \sqsubseteq (\lambda x.N)^{\nu \to \psi}}\ \phi, \psi \notin \mathbf{L}_{\omega}$$

$$(\to \mathrm{E})\ \frac{F^{\phi_1 \wedge \cdots \wedge \phi_n \to \phi} \sqsubseteq G^{\psi_1 \wedge \cdots \wedge \psi_n \to \psi} \qquad \{M^{\phi_1}; \cdots; M^{\phi_m}\} \sqsubseteq \{N^{\psi_1}; \cdots; N^{\psi_n}\}}{(F\{M^{\phi_1}; \cdots; M^{\phi_m}\})^{\phi} \sqsubseteq (G\{N^{\psi_1}; \cdots; N^{\psi_n}\})^{\psi}}$$

$$(\times \mathrm{I})\ \frac{M_1^{\phi_1} \sqsubseteq N_1^{\psi_1} \qquad M_2^{\phi_2} \sqsubseteq N_2^{\psi_2}}{\langle M_1, M_2\rangle^{\phi_1 \times \phi_2} \sqsubseteq \langle N_1, N_2\rangle^{\psi_1 \times \psi_2}}\ \exists i \in \{1,2\}.\phi_i, \psi_i \notin \mathbf{L}_{\omega}$$

$$(\times \mathrm{E}_i)\ \frac{M^{\phi_1 \times \phi_2} \sqsubseteq N^{\psi_1 \times \psi_2}}{(\mathsf{prj}_i M^{\phi_{3-i}})^{\phi_i} \sqsubseteq (\mathsf{prj}_i N^{\psi_{3-i}})^{\psi_i}}\ i \in \{1,2\} \text{ and } \phi_i, \psi_i \notin \mathbf{L}_{\omega}$$

$$(\mathrm{Fix})\ \frac{\{M^{\phi_1}; \cdots; M^{\phi_m}\} \sqsubseteq \{N^{\psi_1}; \cdots; N^{\psi_n}\} \qquad \phi_1 \sqsubseteq \psi_1}{(\mathsf{fix}\, x.\{M^{\phi_1}; \cdots; M^{\phi_m}\})^{\phi_1} \sqsubseteq (\mathsf{fix}\, x.\{N^{\psi_1}; \cdots; N^{\psi_n}\})^{\psi_1}}\ \begin{cases} \forall i \in \{1, \ldots, m\}.\phi_i \notin \mathbf{L}_{\omega} \\ \text{and } \forall j \in \{1, \ldots, n\}.\psi_j \notin \mathbf{L}_{\omega} \end{cases}$$

$$(\mathrm{If})\ \frac{B^{\delta^{\mathsf{nat}}} \sqsubseteq C^{\delta^{\mathsf{nat}}} \qquad M_1^{\phi_1} \sqsubseteq N_1^{\psi_1} \qquad M_2^{\phi_2} \sqsubseteq N_2^{\psi_2} \qquad \phi \sqsubseteq \psi}{(\mathsf{if}\, B\, \mathsf{then}\, M_1^{\phi_1}\, \mathsf{else} M_2^{\phi_2})^{\phi} \sqsubseteq (\mathsf{if}\, C\, \mathsf{then}\, N_1^{\psi_1}\, \mathsf{else} N_2^{\psi_2})^{\psi}}\ \phi, \psi \notin \mathbf{L}_{\omega}$$

$$(\wedge)\ \frac{M^{\phi_{j_1}} \sqsubseteq N^{\psi_1} \ \cdots\ M^{\phi_{j_m}} \sqsubseteq N^{\psi_m}}{\{M^{\phi_1}; \cdots; M^{\phi_n}\} \sqsubseteq \{N^{\psi_1}; \cdots; N^{\psi_m}\}}\ j_1, \ldots, j_m \in \{1, \ldots, n\}$$

Figure 9: Relation $\sqsubseteq$ on $\Lambda^{\vdash}_{\mathbf{T}}(M)$

Let $\sqsubseteq^{\equiv}$ be the relation induced by $\sqsubseteq$ on the equivalence classes. The following property guarantees the existence of optimal faithful $\vdash$-typings, i.e., faithful $\vdash$-typings proving all the useless-code that can be proved by the system $\vdash$.

Theorem 5.19 *For every term M, the set* $\mathbf{faithful}^{\vdash}(M)/\equiv$ *is a finite complete lattice with partial order relation $\sqsubseteq^{\equiv}$ and bottom element $[\delta(\mathrm{FV}(M)) \vdash \delta(M)]^{\equiv}$.*

Optimal faithful $\vdash$-typing can now be defined as follows.

Definition 5.20 (Optimal faithful $\vdash$-typing) *For every PCFP term M we call* optimal faithful $\vdash$-typing of M *every faithful $\vdash$-typing of M whose equivalence class is the top of the lattice* $\mathbf{faithful}^{\vdash}(M)/\equiv$.

For instance, if $\emptyset \vdash Q^{\delta}$, $\emptyset \vdash Q_1^{\delta}$, and $\emptyset \vdash Q_3^{\delta}$ are optimal faithful $\vdash$-typings, then both the faithful $\vdash$-typing $\Sigma \vdash M'$ of Example 5.4 and the faithful $\vdash$-typing $\emptyset \vdash (\mathsf{fix}\, f.\{M^{\phi_1}; M^{\phi_2}\}(\lambda v.v)^{\delta \to \delta} Q_1^{\delta} Q_2^{\omega} Q_3^{\delta})^{\delta}$ of Example 5.16 are optimal.

5.3.2 Optimal simplifications

We can now define a function $\mathbf{O}^{\vdash} : \Lambda_{\mathbf{T}} \to \Lambda_{\mathbf{T}}$ that eliminates from a term M all the useless-code that can detected by $\vdash$. Given M, the function $\mathbf{O}^{\vdash}$ returns $\mathbf{O}(M')$,

where, for some basis Σ, $\Sigma \vdash M'$ is an optimal faithful $\vdash$-typing of M. This mapping is idempotent, i.e., the following property holds.

Theorem 5.21 *For every PCFP term M, $\mathbf{O}^{\vdash}(M) = \mathbf{O}^{\vdash}(\mathbf{O}^{\vdash}(M))$.*

The output of $\mathbf{O}^{\vdash}$ can be further simplified by applying a clean-up procedure, like the one described in [7] (see also Chapter 7 of [5]), which eliminates some of the place-holders d. For instance the simplified term S of Example 5.16 can be replaced by the observationally equivalent term $T = (\mathsf{fix}\, f^{(\mathsf{nat}\to\mathsf{nat})\to\mathsf{nat}\to\mathsf{nat}\to\mathsf{nat}}.U)(\lambda v^{\mathsf{nat}}.v)Q_1Q_3$, where $U = \lambda g^{\mathsf{nat}\to\mathsf{nat}}.\lambda x^{\mathsf{nat}}.\lambda z^{\mathsf{nat}}.\mathsf{ifz}\ z\ \mathsf{then}\ gx\ \mathsf{else}\ f(\lambda u^{\mathsf{nat}}.3)\mathsf{d}_1^{\mathsf{nat}}(-\langle z,1\rangle)$, in which the useless-parameter y and the place-holders d_2 and d_3 have been removed.

Acknowledgments. I thank Stefano Berardi and Mario Coppo for comments and suggestions.

References

[1] S. Berardi. Pruning Simply Typed Lambda Terms. *Journal of Logic and Computation*, 6(5):663–681, 1996.

[2] S. Berardi and L. Boerio. Using Subtyping in Program Optimization. In *TLCA'95*, LNCS 902. Springer–Verlag, 1995.

[3] S. Berardi and L. Boerio. Minimum Information Code in a Pure Functional Language with Data Types. In *TLCA'97*, LNCS 1210. Springer–Verlag, 1997.

[4] M. Coppo, F. Damiani, and P. Giannini. Refinement Types for Program Analysis. In *SAS'96*, LNCS 1145, pages 143–158. Springer–Verlag, 1996.

[5] F. Damiani. *Non-standard type inference for functional programs.* PhD thesis, Dipartimento di Informatica, Università di Torino, February 1998. Available at http://www.di.unito.it/~damiani.

[6] F. Damiani. Useless-code detection and elimination for PCF with algebraic Datatypes. In *TLCA'99*, LNCS 1581, pages 83–97. Springer–Verlag, 1999.

[7] F. Damiani and P. Giannini. Automatic useless-code elimination for HOT functional programs. *Journal of Functional programming.* To appear.

[8] F. Damiani and F. Prost. Detecting and Removing Dead Code using Rank 2 Intersection. In *TYPES'96*, LNCS 1512, pages 66–87. Springer–Verlag, 1998.

[9] A. Fischbach and J. Hannan. Type Systems and Algoritms for Useless-Variable Elimination, 1999. Submitted.

[10] L.S. Hunt and D. Sands. Binding Time Analysis: A New PERspective. In *Partial Evaluation and Semantics-based Program Manipulation.* ACM, 1991.

[11] N. Kobayashi. Type-Based Useless Variable Elimination. In *PEPM'00.* ACM, 2000. To appear.

[12] C. Paulin-Mohring. Extracting F_ω's Programs from Proofs in the Calculus of Constructions. In *POPL'89.* ACM, 1989.

[13] C. Paulin-Mohring. *Extraction de Programme dans le Calcul des Constructions.* PhD thesis, Université Paris VII, 1989.

[14] A. M. Pitts. Operationally-based theories of program equivalence. In A. M. Pitts and P. Dybjer, editors, *Semantics and Logics of Computation*, pages 241–298. Cambridge Univ. Press, 1997.

[15] G. D. Plotkin. LCF considered as a programming language. *Theoretical Computer Science*, 5(3):223–255, 1977.

[16] O. Shivers. *Control Flow Analysis of Higher-Order Languages.* PhD thesis, Carnegie-Mellon University, 1991.

[17] Y. Takayama. Extraction of Redundancy-free Programs from Constructive Natural Deduction Proofs. *Journal of Symbolic Computation*, 12:29–69, 1991.

[18] S. van Bakel. Complete restrictions of the intersection type discipline. *Theoretical Computer Science*, 102:135–163, 1992.

[19] M. Wand and I. Siveroni. Constraints Systems for Useless Variable Elimination. In *POPL'99*, pages 291–302. ACM, 1999.

A Proofs

Ground contextual equivalence admits a co-inductive characterization in terms of bisimulation (see, for instance, [14]). We give here the definition of bisimulation for PCFP, and state two results that will be used in the following.

Definition A.1 (PCFP bisimulations and bisimilarity) *1. A* PCFP bisimulation $\mathcal{B}$ *is a type indexed family of relations on closed terms,*

$$\mathcal{B}_\rho \subseteq \{\langle M, N\rangle \mid M, N \in \rho \text{ and } \mathrm{FV}(M) \cup \mathrm{FV}(N) = \emptyset\}$$

($\rho \in \mathbf{T}$), satisfying the conditions in Fig. 10. PCFP bisimilarity *is the largest bisimulation and will be denoted by $\simeq$.*

2. A PCFP bisimulation $\mathcal{B}$ can be extended to a relation on (possibly) open terms $\mathcal{B}^{\mathrm{o}}$ (called the open extension *of $\mathcal{B}$). For any PCFP terms P and Q such that $\mathrm{FV}(P) \cup \mathrm{FV}(Q) \subseteq \{x_1^{\sigma_1}, \ldots, x_n^{\sigma_n}\}$, define: $M\ \mathcal{B}^{\mathrm{o}}_\rho\ N$ if and only if, for every closed substitution $[x_1^{\sigma_1} := N_1, \ldots, x_n^{\sigma_n} := N_n]$,*

$$P[\vec{x} := \vec{N}]\ \mathcal{B}_\rho\ Q[\vec{x} := \vec{N}].$$

Let $\simeq^{\mathrm{o}}$ denote the PCFP open bisimilarity*, i.e., the relation obtained by applying the previous construction to $\simeq$.*

(bis 1a)	$(M\ \mathcal{B}_{\mathsf{nat}}\ N$ and $M \Downarrow \mathsf{n})$ implies $N \Downarrow \mathsf{n}$
(bis 1b)	$(M\ \mathcal{B}_{\mathsf{nat}}\ N$ and $N \Downarrow \mathsf{n})$ implies $M \Downarrow \mathsf{n}$
(bis 2)	$M\ \mathcal{B}_{\rho\to\sigma}\ N$ implies, for all P such that $P \in \Lambda_\rho$, $MP\ \mathcal{B}_\sigma\ NP$
(bis 3)	$M\ \mathcal{B}_{\rho\times\sigma}\ N$ implies $(\mathsf{prj}_1(M)\ \mathcal{B}_\rho\ \mathsf{prj}_1(N)$ and $\mathsf{prj}_2(M)\ \mathcal{B}_\sigma\ \mathsf{prj}_2(N))$

Figure 10: Bisimulation conditions for PCFP

The following results hold.

Theorem A.2 (Operational extensionality for PCFP) *Ground contextual equivalence coincides with bisimilarity, that is:*

$$M \simeq_{\mathrm{obs}} N \text{ if and only if } M \simeq^{\mathrm{o}} N.$$

Theorem A.3 (A "compactness" property of evaluation) *For each natural number n, define $\mathsf{fix}^{(n)}\ x.M$ as follows:*

$$\begin{array}{lcl} \mathsf{fix}^{(0)}\ x.M & = & \mathsf{fix}\ z.z \\ \mathsf{fix}^{(n+1)}\ x.M & = & M[x := \mathsf{fix}^{(n)}\ x.M]\ . \end{array}$$

For every PCFP context of type nat, $(\mathcal{C}[\,]^\rho)^{\mathsf{nat}}$, *if* $(\mathcal{C}[\mathsf{fix}\ x.M]^\rho)^{\mathsf{nat}} \Downarrow \mathsf{k}$ *then exists $m \geq 0$ such that* $(\mathcal{C}[\mathsf{fix}^{(m)}\ x.M]^\rho)^{\mathsf{nat}} \Downarrow \mathsf{k}$.

A.1 From Section 4.1

Restatement of Proposition 4.4.

1. $\phi^\rho \in \mathbf{L}_\omega$ *(i.e.* $\phi = \omega^\rho$*) if and only if* $[\![\phi^\rho]\!] = \Omega^\rho$*, and*
2. $\phi^\rho \in \mathbf{L}_\delta$ *implies* $[\![\phi^\rho]\!] = \Delta^\rho$.

Proof.

1. ($\Longrightarrow$). Immediate from Definition 4.2.
   ($\Longleftarrow$). Let $[\![\phi^\rho]\!] = \Omega^\rho$. By induction on the structure of e-types.

   $\phi^\rho \in \{\delta^{\mathsf{nat}}, \omega^\rho\}$. Immediate from Definition 4.2.

   $\phi^\rho = \xi^{\rho_1} \to \psi^{\rho_2}$. By Definition 4.1, $\psi \neq \omega^{\rho_2}$. For all $[P], [Q] \in \mathbf{I}(\rho_2)$, $([\lambda z^{\rho_1}.P], [\lambda z^{\rho_1}.Q]) \in [\![\xi \to \psi]\!]$. So from Definition 4.2 we get $[\![\psi]\!] = \Omega^{\rho_2}$, and (by induction) $\psi = \omega^{\rho_2}$, which is absurd.

   $\phi^\rho = {\phi_1}^{\rho_1} \times {\phi_2}^{\rho_2}$. By Definition 4.1, ${\phi_i}^{\rho_i} \neq \omega^{\rho_i}$, for some $i \in \{1, 2\}$. From Definition 4.2, $[\![\phi^\rho]\!] = \Omega^\rho$ implies both $[\![\phi_1]\!] = \Omega^{\rho_1}$ and $[\![\phi_2]\!] = \Omega^{\rho_2}$. So (by induction hypothesis) $\phi_1, \phi_2 \in \mathbf{L}_\omega$, which is absurd.

2. We prove that, for all $\rho \in \mathbf{T}$, $[\![\delta(\rho)]\!] = \Delta^\rho$. By induction on the structure of types.

   $\rho = \mathsf{nat}$. Immediate.

   $\rho = \rho_1 \to \rho_2$. By induction hypothesis both $[\![\delta(\rho_1)]\!] = \Delta^{\rho_1}$ and $[\![\delta(\rho_2)]\!] = \Delta^{\rho_2}$. By definition of $[\![\delta(\rho_1 \to \rho_2)]\!]$ we have that

   (∗) $([P], [Q] \in [\![\delta(\rho_1 \to \rho_2)]\!]$ if and only if, for all M and N such that $([M], [N]) \in [\![\delta(\rho_1)]\!]$, $([P\,M], [Q\,N]) \in [\![\delta(\rho_2)]\!]$.

   We first show that

   (a) $([P], [Q]) \in [\![\delta(\rho_1 \to \rho_2)]\!]$ implies $P \simeq Q$, and
   (b) $P \simeq Q$ implies $([P], [Q]) \in [\![\delta(\rho_1 \to \rho_2)]\!]$.

   The first implication is proved by absurd. If $P \not\simeq Q$, then, by definition of bisimilarity there would be an M such that $P\,M \not\simeq Q\,M$. So (by induction hypothesis) $([P\,M], [Q\,M]) \notin [\![\delta(\rho_2)]\!]$, and therefore (from (∗)) $([P], [Q]) \notin [\![\delta(\rho_1 \to \rho_2)]\!]$. For the second one, let $P \simeq Q$. This implies that, for all R and S such that $R \simeq S$, $P\,R \simeq P\,S \simeq Q\,S \simeq Q\,R$. So (by induction hypothesis and (∗)) $([P], [Q]) \in [\![\delta(\rho_1 \to \rho_2)]\!]$.
   Therefore from the equivalence between $\simeq$ and contextual equivalence (Theorem A.2) we get the result.

   $\rho = \rho_1 \times \rho_2$. Similar.

A.2 From Section 4.2

Restatement of Theorem 4.6. $\xi_1 \leq \xi_2$ *implies* $[\![\xi_1]\!] \subseteq [\![\xi_2]\!]$.

Proof. By induction on the structure of derivations. □

A.3 From Section 5.1

A useful property of the e-types semantics is the following.

Proposition A.4 *If, for all* $n \geq 0$, $([(\mathcal{C}_1[\mathsf{fix}^{(n)}\, x.M]^\rho)^\sigma], [(\mathcal{C}_2[\mathsf{fix}^{(n)}\, x.M]^\rho)^\sigma]) \in [\![\phi^\sigma]\!]$, *then* $([(\mathcal{C}_1[\mathsf{fix}\, x.M]^\rho)^\sigma], [(\mathcal{C}_2[\mathsf{fix}\, x.M]^\rho)^\sigma]) \in [\![\phi^\sigma]\!]$.

Proof. By induction on the structure of e-types.

$\phi^\sigma = \omega^\sigma$. Immediate.

$\phi^\sigma = \delta^{\mathsf{nat}}$. By Theorem A.3.

$\phi^\sigma = \xi^{\sigma_1} \to \psi^{\sigma_2}$. By contraposition assume that $([(\mathcal{C}_1[\mathsf{fix}\, x.M]^\rho)^\sigma], [(\mathcal{C}_2[\mathsf{fix}\, x.M]^\rho)^\sigma]) \notin [\![\phi^\sigma]\!]$. Then, for some $([P], [Q]) \in [\![\xi^{\sigma_1}]\!]$, we have

$$([(\mathcal{C}_1[\mathsf{fix}\, x.M]^\rho)^\sigma P], [(\mathcal{C}_2[\mathsf{fix}\, x.M]^\rho)^\sigma Q]) \notin [\![\psi^{\sigma_2}]\!].$$

By induction hypothesis there exists $m \geq 0$ such that

$$([(\mathcal{C}_1[\mathsf{fix}^{(m)}\, x.M]^\rho)^{\mathsf{nat}} P], [(\mathcal{C}_2[\mathsf{fix}^{(m)}\, x.M]^\rho)^{\mathsf{nat}} Q]) \notin [\![\psi^{\sigma_2}]\!].$$

Therefore $([(\mathcal{C}_1[\mathsf{fix}^{(m)}\, x.M]^\rho)^{\mathsf{nat}}], [(\mathcal{C}_2[\mathsf{fix}^{(m)}\, x.M]^\rho)^{\mathsf{nat}}]) \notin [\![\phi^\sigma]\!]$.

$\phi^\sigma = \phi_1^{\sigma_1} \times \phi_2^{\sigma_2}$. Similar.

□

Restatement of Theorem 5.3 *Let* $\Sigma \vdash P^\chi$. *Then* $\epsilon(P^\chi) \sim_\chi^\Sigma \epsilon(P^\chi)$.

Proof. By induction on the structure of derivations. Let e and e' be two Σ-related environments. We only present four cases.

- Let the derivation end with rule (Var). So $P = x$. If e and e' are Σ-related then $(e(x), e'(x)) \in [\![\phi_1 \wedge \cdots \wedge \phi_n]\!]$. By Theorem 4.6 (soundness of $\leq$), $[\![\phi_1 \wedge \cdots \wedge \phi_n]\!] \subseteq [\![\chi]\!]$, so $(e(x), e'(x)) \in [\![\chi]\!]$ which proves the result.

- Let the derivation end with the rule ($\to$ I). So $P = \lambda x.M$ and $\chi = \xi \to \psi$. We have to prove that $([\![\lambda x.M]\!]_e, [\![\lambda x.M]\!]_{e'}) \in [\![\xi \to \psi]\!]$, i.e. $([(\lambda x.M)[\bar{y} := \bar{Q}]], [(\lambda x.M)[\bar{y} := \bar{Q}']]) \in [\![\xi \to \psi]\!]$, where $\bar{y} = y_1 \cdots y_k$ ($k \geq 0$) are the variables in Σ, $e(y_i) = [Q_i]$, $e'(y_i) = [Q_i']$, ξ_i is the conjunction assumed for y_i in Σ, and $(e(y_i), e'(y_i)) \in [\![\xi_i]\!]$. By definition this amounts to show that, for all $([N], [N']) \in [\![\xi]\!]$, $([(\lambda x.M)[\bar{y} := \bar{Q}]N], [(\lambda x.M)[\bar{y} := \bar{Q}']N']) \in [\![\psi]\!]$, i.e., since $\mathrm{FV}(N) = \mathrm{FV}(N') = \emptyset$, $([M[x := N, \bar{y} := \bar{Q}]], [M[x := N', \bar{y} := \bar{Q}']]) \in [\![\psi]\!]$, that is $([\![M]\!]_{e[x:=[N]]}, [\![M]\!]_{e[x:=[N]]}) \in [\![\psi]\!]$, which follows directly by induction.

- Let the derivation end with the rule ($\to$ E). So $P = MN$ and $\chi = \psi$. We have to prove that $([\![MN]\!]_e, [\![MN]\!]_{e'}) \in [\![\psi]\!]$, i.e. $([(MN)[\bar{y} := \bar{Q}]], [(MN)[\bar{y} := \bar{Q}']]) \in [\![\psi]\!]$, where $\bar{y} = y_1 \cdots y_k$ ($k \geq 0$) are the variables in Σ, $e(y_i) = [Q_i]$, $e'(y_i) = [Q_i']$, ξ_i is the conjunction assumed for y_i in Σ, and $(e(y_i), e'(y_i)) \in [\![\xi_i]\!]$. By induction we have $([M[\bar{y} := \bar{Q}]], [M[\bar{y} := \bar{Q}']]) \in [\![\phi_1 \wedge \cdots \wedge \phi_n \to \psi]\!]$ and, for all $j \in \{1, \ldots, n\}$, $([N[\bar{y} := \bar{Q}]], [N[\bar{y} := \bar{Q}']]) \in [\![\phi_j]\!]$. So $([M[\bar{y} := \bar{Q}]\, N[\bar{y} := \bar{Q}]], [M[\bar{y} := \bar{Q}']\, N[\bar{y} := \bar{Q}']]) \in [\![\psi]\!]$, i.e., $([MN[\bar{y} := \bar{Q}]], [MN[\bar{y} := \bar{Q}']]) \in [\![\psi]\!]$.

- Let the derivation end with the rule (Fix). So $P = \mathsf{fix}\, x.M$ and $\chi = \phi_1$. We first prove that, for every $j \in \{1, \ldots, n\}$, $(\llbracket \mathsf{fix}\, x.M \rrbracket_e, \llbracket \mathsf{fix}\, x.M \rrbracket_{e'}) \in \llbracket \phi_j \rrbracket$, i.e. $([(\mathsf{fix}\, x.M)[\bar{y} := \bar{Q}]], [(\mathsf{fix}\, x.M)[\bar{y} := \bar{Q}']]) \in \llbracket \phi_j \rrbracket$, where $\bar{y} = y_1 \cdots y_k$ $(k \geq 0)$ are the variables in Σ, $e(y_i) = [Q_i]$, $e'(y_i) = [Q_i']$, ξ_i is the conjunction assumed for y_i in Σ, and $(e(y_i), e'(y_i)) \in \llbracket \xi_i \rrbracket$.

  Since $\mathsf{fix}^{(0)}\, x.P = \mathsf{fix}\, z.z$ for any P, we have $([\mathsf{fix}^{(0)}\, x.M], [\mathsf{fix}^{(0)}\, x.M]) \in \llbracket \phi_j \rrbracket$. Assuming that $(\llbracket \mathsf{fix}^{(n)}\, x.M \rrbracket_e, \llbracket \mathsf{fix}^{(n)}\, x.M \rrbracket_{e'}) \in \llbracket \phi_j \rrbracket$ we get (by induction from the derivations $\Sigma, x : \phi_1 \wedge \cdots \wedge \phi_n \vdash M : \phi_j$) $(\llbracket M \rrbracket_{e[x := \llbracket \mathsf{fix}^{(n)}\, x.M \rrbracket_e]}, \llbracket M \rrbracket_{e'[x := \llbracket \mathsf{fix}^{(n)}\, x.M \rrbracket_{e'}]}) \in \llbracket \phi_j \rrbracket$, i.e. $([M[x := (\mathsf{fix}^{(n)}\, x.M)[\bar{y} := \bar{Q}], \bar{y} := \bar{Q}]], [M[x := (\mathsf{fix}^{(n)}\, x.M)[\bar{y} := \bar{Q}], \bar{y} := \bar{Q}]]) \in \llbracket \phi_j \rrbracket$, i.e. $([(\mathsf{fix}^{(n+1)}\, x.M)[\bar{y} := \bar{Q}]], [(\mathsf{fix}^{(n+1)}\, x.M)[\bar{y} := \bar{Q}]]) \in \llbracket \phi_j \rrbracket$, i.e. $(\llbracket \mathsf{fix}^{(n+1)}\, x.M \rrbracket_e, \llbracket \mathsf{fix}^{(n+1)}\, x.M \rrbracket_{e'}) \in \llbracket \phi_j \rrbracket$. Then, by induction on n, we have that for all n, $(\llbracket \mathsf{fix}^{(n)}\, x.M \rrbracket_e, \llbracket \mathsf{fix}^{(n)}\, x.M \rrbracket_{e'}) \in \llbracket \phi_j \rrbracket$. So we can conclude by Proposition A.4 that $(\llbracket \mathsf{fix}\, x.M \rrbracket_e, \llbracket \mathsf{fix}\, x.M \rrbracket_{e'}) \in \llbracket \phi_j \rrbracket$. □

Restatement of Lemma 5.6 *Let $\Sigma_1 \vdash M'^{\delta(\rho)}$ and $\Sigma_2 \vdash N'^{\delta(\rho)}$ be faithful $\vdash$-typings (of terms M and N, respectively) such that Σ_1 and Σ_2 contains the same assumptions for the variables in $\mathrm{FV}(M) \cap \mathrm{FV}(N)$. Then $M \sim_\phi^{\Sigma_1 \cup \Sigma_2} N$ implies $M \simeq_{\mathrm{obs}} N$.*

Proof. Observe that $\Sigma_1 \cup \Sigma_2$ contains exactly the free variables of M and N. Since $\Sigma_1 \vdash M'^{\phi}$ and $\Sigma_2 \vdash N'^{\phi}$ are faithful, for any environment e, e is $\Sigma_1 \cup \Sigma_2$-related with e. Let now $M \sim_\phi^{\Sigma_1 \cup \Sigma_2} N$, then (by Definition 5.2.2) $\langle \llbracket M \rrbracket_e, \llbracket N \rrbracket_e \rangle \in \llbracket \delta(\rho) \rrbracket$, i.e., $\llbracket M \rrbracket_e = \llbracket N \rrbracket_e$. Using the definition of $\simeq^{\circ}$ and Theorem A.2 we can conclude that $M \simeq_{\mathrm{obs}} N$. □

A.4 From Section 5.2

To simplify the presentation of some of the proofs that we will develop in the following, it is useful to consider the inductive characterizations of the pruning relation $\preceq_{\mathrm{prn}}$ of Definition 5.8 and of the operation **inf** of Definition 5.10: the relation $\preceq_{\mathrm{prn}}$ is defined by the clauses in Fig. 11, and the operation **inf** is defined by the clauses in Fig. 12.

Restatement of Theorem 5.11 *For every PCFP term M of type ρ, the set $\Lambda_{\mathbf{T}}^{\preceq_{\mathrm{prn}}}(M)$ is a finite complete lattice with order relation $\preceq_{\mathrm{prn}}$, meet operation* **inf**, *bottom element M, and top element d^ρ.*

Proof. It is immediate to verify that $\preceq_{\mathrm{prn}}$ is a partial order relation on the finite set $\Lambda_{\mathbf{T}}^{\preceq_{\mathrm{prn}}}(M)$ (remember that we identify prunings of M modulo renaming of the dummy variables), and that M is the bottom and d^ρ is the top. To complete the proof we have only to show that, for every $M_1, M_2 \in \Lambda_{\mathbf{T}}^{\preceq_{\mathrm{prn}}}(M)$, the term $\mathbf{inf}(M_1, M_2)$ is the greatest lower bound of M_1 and M_2 in $\Lambda_{\mathbf{T}}^{\preceq_{\mathrm{prn}}}(M)$. This can be proved by induction on the structure of terms, following the inductive definition of **inf** (see Fig. 12). □

Restatement of Lemma 5.13 *For every term M, if $\Sigma \vdash M'^{\psi}$ is a $\vdash$-typing of M and N is term such that $M \preceq_{\mathrm{prn}} N \preceq_{\mathrm{prn}} \mathbf{O}(M'^{\psi})$, then*

$M \preceq_{\mathrm{prn}} \mathsf{d}^\rho$, for every term M of type ρ and fresh dummy variable d^ρ

$x^\rho \preceq_{\mathrm{prn}} x^\rho$

$\mathsf{k}^{\mathbf{T}(\mathsf{k})} \preceq_{\mathrm{prn}} \mathsf{k}^{\mathbf{T}(\mathsf{k})}$

$\lambda x^\sigma.M \preceq_{\mathrm{prn}} \lambda y^\sigma.N$, if $M[x^\sigma := z^\sigma] \preceq_{\mathrm{prn}} N[y^\sigma := z^\sigma]$ for some fresh variable z^σ

$FM \preceq_{\mathrm{prn}} GN$, if $F \preceq_{\mathrm{prn}} G$ and $M \preceq_{\mathrm{prn}} N$

$\langle M_1, M_2\rangle \preceq_{\mathrm{prn}} \langle N_1, N_2\rangle$, if, for $i \in \{1,2\}, M_i \preceq_{\mathrm{prn}} N_i$

$\mathsf{prj}_i M \preceq_{\mathrm{prn}} \mathsf{prj}_i N$, if $M \preceq_{\mathrm{prn}} N$

$\mathsf{fix}\, x^\sigma.M \preceq_{\mathrm{prn}} \mathsf{fix}\, y^\sigma.N$, if $M[x^\sigma := z^\sigma] \preceq_{\mathrm{prn}} N[y^\sigma := z^\sigma]$ for some fresh variable z^σ

$\mathsf{ifz}\, B\, \mathsf{then}\, M_1\, \mathsf{else} M_2 \preceq_{\mathrm{prn}} \mathsf{ifz}\, C\, \mathsf{then}\, N_1\, \mathsf{else} N_2$, if $B \preceq_{\mathrm{prn}} C$ and, for $i \in \{1,2\}, M_i \preceq_{\mathrm{prn}} N_i$

Figure 11: Relation $\preceq_{\mathrm{prn}}$ on $\Lambda_{\mathbf{T}}$

1. $\Sigma' \vdash N'^{\psi}$, *where* $\Sigma' = \{x : \xi \in \Sigma \mid x \in \mathrm{FNDV}(N)\} \cup \{\mathsf{d}^\sigma : \omega^\sigma \mid \mathsf{d}^\sigma \in \mathrm{DV}(N)\}$ *and* N'^{ψ} *is the decoration of* N *obtained from* M'^{ψ} *by replacing by* d *the subterms which have been replaced by* d *in* N.
2. $M \sim_{\psi}^{\Sigma''} N$, *where* $\Sigma'' = \Sigma \cup \{\mathsf{d}^\sigma : \omega^\sigma \mid \mathsf{d}^\sigma \in \mathrm{DV}(N)\}$.

Proof.

1. By definition of $\mathbf{O}$, every dummy variable in $\mathbf{O}(M'^{\psi})$ corresponds to a subexpression of M which, in M'^{ψ}, is always decorated with an ω-e-type (and vice versa). This implies that any dummy variable in N must occur in the scope of a subexpression of M that, in M'^{ψ}, is always decorated with an ω-e-type. So we have that $\Sigma \vdash M'^{\psi}$ implies $\Sigma' \vdash N'^{\psi}$.
2. By induction on the structure of derivations, observing that $\Sigma \vdash M'^{\psi}$ implies both $\Sigma'' \vdash M'^{\psi}$ and $\Sigma'' \vdash N'^{\psi}$.

□

Restatement of Theorem 5.14 *Let* $\Sigma \vdash M^{\delta(\rho)}$ *be a faithful* $\vdash$*-typing. Then* $\epsilon(M^{\delta(\rho)}) \simeq_{\mathrm{obs}} \mathbf{O}(M^{\delta(\rho)})$.

Proof. Let $\Sigma' = \{x^\sigma : \xi \in \Sigma \mid x^\sigma \in \mathrm{FNDV}(\mathbf{O}(M^{\delta(\rho)}))\} \cup \{\mathsf{d}^\sigma : \omega^\sigma \mid \mathsf{d}^\sigma \in \mathrm{DV}(\mathbf{O}(M^{\delta(\rho)}))\}$. Since $\Sigma \vdash M^{\delta(\rho)}$ is a faithful $\vdash$-typing then also $\Sigma' \vdash P^{\delta(\rho)}$ (where $P^{\delta(\rho)}$ is the decoration of $\mathbf{O}(M^{\delta(\rho)})$ obtained from $M^{\delta(\rho)}$ by replacing by d the subterms which have been replaced by d in $\mathbf{O}(M^{\delta(\rho)})$) is a faithful $\vdash$-typing. So the result follows immediately from Lemma 5.13 and Lemma 5.6. □

$$\mathbf{inf}(M, \mathsf{d}^{\rho}) = \mathbf{inf}(\mathsf{d}^{\rho}, M) = M, \text{ for every term } M \text{ of type } \rho$$

$$\mathbf{inf}(\mathsf{k}^{\rho}, \mathsf{k}^{\rho}) = \mathsf{k}^{\rho}$$

$$\mathbf{inf}(x^{\rho}, x^{\rho}) = x^{\rho}$$

$$\mathbf{inf}(\langle M_1, M_2\rangle, \langle N_1, N_2\rangle) = \langle \mathbf{inf}(M_1, N_1), \mathbf{inf}(M_2, N_2)\rangle$$

$$\mathbf{inf}(\mathsf{prj}_i M, \mathsf{prj}_i N) = \mathsf{prj}_i \mathbf{inf}(M, N), \quad \text{where } i \in \{1, 2\}$$

$$\mathbf{inf}(M_1 M_2, N_1 N_2) = \mathbf{inf}(M_1, N_1)\mathbf{inf}(M_2, N_2)$$

$$\mathbf{inf}(\lambda x^{\rho}.M, \lambda x^{\rho}.N) = \lambda x^{\rho}.\mathbf{inf}(M, N)$$

$$\mathbf{inf}(\mathsf{fix}\, x^{\rho}.M, \mathsf{fix}\, x^{\rho}.N) = \mathsf{fix}\, x^{\rho}.\mathbf{inf}(M, N)$$

$$\mathbf{inf}(\mathsf{ifz}\, M\, \mathsf{then}\, M_1\, \mathsf{else}\, M_2,\ \mathsf{ifz}\, N\, \mathsf{then}\, N_1\, \mathsf{else}\, N_2) = \mathsf{ifz}\, \mathbf{inf}(M, N)\, \mathsf{then}\, \mathbf{inf}(M_1, N_1)\, \mathsf{else}\, \mathbf{inf}(M_2, N_2)$$

Figure 12: Operation **inf** on $\Lambda^{\preceq_{\mathrm{prn}}}_{\mathbf{T}}(M)$

A.5 From Section 5.3

Restatement of Lemma 5.18 *For every term P, if $\Sigma_1 \vdash P_1{}^{\chi_1}$ and $\Sigma_2 \vdash P_2{}^{\chi_2}$ two $\vdash$-typings of P such that $\Sigma_1 \vdash P_1{}^{\chi_1} \sqsubseteq \Sigma_2 \vdash P_2{}^{\chi_2}$, then $\mathbf{O}(P_1{}^{\chi_1}) \preceq_{\mathrm{prn}} \mathbf{O}(P_2{}^{\chi_2})$.*

Proof. By structural induction on the structure of $\sqsubseteq$-derivations. If the derivation of $\Sigma_1 \vdash P_1{}^{\chi_1} \sqsubseteq \Sigma_2 \vdash P_2{}^{\chi_2}$ ends by rule (ω), (Var), or (Con) the proof is immediate. The remaining cases are direct by induction. We consider only the case of rule (Fix). In such case, we have $P_1 = \mathsf{fix}\, x.\{M^{\phi_1}; \cdots; M^{\phi_m}\}$, $P_2 = \mathsf{fix}\, x.\{N^{\psi_1}; \cdots; N^{\psi_n}\}$, $\chi_1 = \phi_1$, and $\chi_2 = \psi_1$. By induction, for some $j_1, \ldots, j_m \in \{1, \ldots, n\}$ and for all $h \in \{1, \ldots, m\}$, $\mathbf{O}(M^{\phi_{j_h}}) \preceq_{\mathrm{prn}} \mathbf{O}(N^{\psi_h})$. So, by definition of $\preceq_{\mathrm{prn}}$, $\mathsf{fix}\, x.\mathbf{inf}(\mathbf{O}(M^{\phi_1}), \ldots, \mathbf{O}(M^{\phi_m})) \preceq_{\mathrm{prn}} \mathsf{fix}\, x.\mathbf{inf}(\mathbf{O}(N^{\psi_1}), \ldots, \mathbf{O}(N^{\psi_n}))$. □

In order to prove the existence of optimal faithful $\vdash$-typings for every PCFP term M, we define an operation, $\sqcup$, on the set of the $\vdash$-typings of M which allows to "combine" two faithful $\vdash$-typings of M ($\Sigma_1 \vdash M_1^{\delta(\rho)}$ and $\Sigma_2 \vdash M_2^{\delta(\rho)}$) to produce a faithful $\vdash$-typing of M ($\sqcup(\Sigma_1 \vdash M_1^{\delta(\rho)}, \Sigma_2 \vdash M_2^{\delta(\rho)})$) which proves the useless-code proved by *either one or the other* of the $\vdash$-typings. For every PCFP term M of type ρ and for every finite set of variables Γ such that $\Gamma \supseteq \mathrm{FV}(M)$, consider the sets:

$$\begin{aligned} \mathbf{bases}^{\vdash}(\Gamma) &= \{\Sigma \mid \Sigma \text{ is a basis and } \epsilon(\Sigma) = \Gamma\}\ , \\ \mathbf{typings}^{\vdash}(M, \Gamma) &= \{\Sigma \vdash M'^{\phi} \mid \Sigma \vdash M'^{\phi} \text{ is a } \vdash\text{-typing of } M \text{ and } \epsilon(\Sigma) = \Gamma\}\ . \end{aligned}$$

Definition A.5 (Operation $\sqcup$) *1. For every type ρ the binary operation $\sqcup$ on the set $\mathbf{L}(\rho)$ is defined by the rules in Fig. 13.*

2. *For every set of term variables* Γ *the binary operation* $\sqcup$ *on the set* $\mathbf{bases}^{\vdash}(\Gamma)$ *is defined by:* $\sqcup(\Sigma_1, \Sigma_2) = \{x^{\rho} : \sqcup(\nu_1, \nu_2) \mid x^{\rho} : \nu_1 \in \Sigma_1 \text{ and } x^{\rho} : \nu_1 \in \Sigma_2\}$.

3. *For every term* M *the binary operation* $\sqcup$ *on the set of* $\vdash$*-decorated terms* $\Lambda^{\vdash}_{\mathbf{T}}(M)$ *is defined by the rules in Fig. 14.*

4. *For every term* M *and for every set of term variables* Γ *such that* $\Gamma \supseteq \mathrm{FV}(M)$*, the binary operation* $\sqcup$ *on the set* $\mathbf{typings}^{\vdash}(M, \Gamma)$ *is defined by:* $\sqcup(\Sigma_1 \vdash M_1^{\phi_1},\ \Sigma_2 \vdash M_2^{\phi_2}) = \sqcup(\Sigma_1, \Sigma_2) \vdash \sqcup(M_1^{\phi_1},\ M_2^{\phi_2})$.

It is immediate to verify that, for every PCFP type $\rho \in \mathbf{T}$ and for every $\phi_1, \phi_2 \in \mathbf{L}(\rho)$, we have that $\sqcup(\phi_1, \phi_2) \in \mathbf{L}(\rho)$. Similarly for basis. The fact that the operation $\sqcup$ is well defined also on $\vdash$-decorated terms and on $\vdash$-typings is guaranteed by the following lemmas.

Lemma A.6 *For every PCFP type* $\rho \in \mathbf{T}$ *and for all conjunctions* ξ, ξ', ν, ν' *of underlying type* ρ*. If* $\xi \leq \xi'$ *and* $\nu \leq \nu'$ *then* $\sqcup(\xi, \nu) \leq \sqcup(\xi', \nu')$*.*

Proof. By structural induction on derivations. If either the derivation of $\xi \leq \xi'$ or the derivation of $\nu \leq \nu'$ consists in an application of rule (ω), the proof is immediate. If both the derivations end by the same rule, then the result is straightforward by induction. Let us now consider the cases in which the derivations end by different rules.

- Let the first derivation end by rule (δ). Then the second derivation can only end by rule $(\wedge)$ and all the premise's derivations must end by either rule (δ) or rule (ω). So the result is immediate.
- Let the first derivation end by rule $(\to)$. Then the second derivation can only end by rule $(\wedge)$ and all the premise's derivations must end by either rule $(\to)$ or rule (ω). So the result is straightforward by induction.
- Let the first derivation end by rule $(\times)$. Then the second derivation can only end by rule $(\wedge)$ and all the premise's derivations must end by either rule $(\times)$ (with 1 premise) or rule (ω). So the result is straightforward by induction.
- The remaining cases are symmetric.

□

Lemma A.7 *For every PCFP term* M*, if* $\Sigma_1 \vdash M_1^{\phi_1}$ *and* $\Sigma_2 \vdash M_2^{\phi_2}$ *are* $\vdash$*-typings of* M *such that* $\epsilon(\Sigma_1) = \epsilon(\Sigma_2)$ *then* $\sqcup(\Sigma_1, \Sigma_2) \vdash \sqcup(M_1^{\phi_1},\ M_2^{\phi_2})$*.*

Proof. By structural induction on derivations. If one of the two derivations $\Sigma_1 \vdash M_1^{\phi_1}$ and $\Sigma_2 \vdash M_2^{\phi_2}$ consists in an application of rule (ω) the proof is immediate. Otherwise both the derivations must end by the same rule, and the result is straightforward by induction (using Lemma A.6 for rules (Var) and (If)). □

The following lemmas show that the operation $\sqcup$ is the least-upper bound operator w.r.t. the partial order $\sqsubseteq$.

$$\sqcup(\delta^{\mathsf{nat}}, \delta^{\mathsf{nat}}) = \delta^{\mathsf{nat}}$$

$$\sqcup(\phi^\rho, \omega^\rho) = \sqcup(\omega^\rho, \phi^\rho) = \omega^\rho$$

$$\sqcup(\xi \to \phi, \nu \to \psi) = \sqcup(\xi, \nu) \to \sqcup(\phi, \psi)$$

$$\sqcup(\phi_1 \times \phi_2, \psi_1 \times \psi_2) = \sqcup(\phi_1, \psi_1) \times \sqcup(\phi_2, \psi_2)$$

$$\sqcup(\phi_1 \wedge \cdots \wedge \phi_n, \psi_1 \wedge \cdots \wedge \psi_m) = \wedge_{i \in \{1,\ldots,n\}, j \in \{1,\ldots,m\}} \sqcup(\phi_i, \psi_j)$$

Figure 13: Operation $\sqcup$ on $\mathbf{L}(\rho)$

Lemma A.8 *For every PCFP type $\rho \in \mathbf{T}$ and for all conjunctions ξ_1, ξ_2, ξ_3 of underlying type ρ.*

1. *$\xi_1 \sqsubseteq \sqcup(\xi_1, \xi_2)$ and $\xi_2 \sqsubseteq \sqcup(\xi_1, \xi_2)$.*
2. *If $\xi_1 \sqsubseteq \xi_3$ and $\xi_2 \sqsubseteq \xi_3$ then $\sqcup(\xi_1, \xi_2) \sqsubseteq \xi_3$.*

Proof. Both point 1 and point 2 by structural induction on $\sqsubseteq$-derivations. We consider only point 2.

If either the derivation of $\xi_1 \sqsubseteq \xi_3$ or the derivation of $\xi_2 \sqsubseteq \xi_3$ consists in an application of rule (ω), the proof is immediate. If both the derivations end by the same rule, then the result is straightforward by induction: we consider only the case of rule $(\wedge)$. In such case, we have $\xi_1 = \phi_1 \wedge \cdots \wedge \phi_m$, $\xi_2 = \psi_1 \wedge \cdots \wedge \psi_n$, and $\xi_3 = \eta_1 \wedge \cdots \wedge \eta_q$. By definition of $\sqsubseteq$, for all $h \in \{1, \ldots, q\}$,

- for some $j_1, \ldots, j_q \in \{1, \ldots, n\}$, $\phi_{j_h} \sqsubseteq \eta_h$,
- for some $i_1, \ldots, i_q \in \{1, \ldots, m\}$, $\psi_{i_h} \sqsubseteq \eta_h$.

So, by induction, we have that, for all $h \in \{1, \ldots, q\}$,

$$\sqcup(\phi_{j_h}, \psi_{i_h}) \sqsubseteq \eta_h,$$

which, by rule $(\wedge)$, gives the result.

Let us now consider the cases in which the derivations end by different rules.

- Let the first derivation end by rule (δ). Then the second derivation can only end by rule $(\wedge)$ and all the premise's derivations must end by rule (δ). So the result is straightforward by induction.
- Let the first derivation end by rule $(\to)$. Then the second derivation can only end by rule $(\wedge)$ and all the premise's derivations must end by rule $(\to)$. So the result is straightforward by induction.
- Let the first derivation end by rule $(\times)$. Then the second derivation can only end by rule $(\wedge)$ and all the premise's derivations must end by rule $(\times)$. So the result is straightforward by induction.

$\sqcup(M^{\phi}, \epsilon(M)^{\omega^{\epsilon(\phi)}}) = \sqcup(\epsilon(M)^{\omega^{\epsilon(\phi)}}, M^{\phi}) = \epsilon(M)^{\omega^{\epsilon(\phi)}}$

$\sqcup(x^{\phi}, x^{\psi}) = x^{\sqcup(\phi,\psi)}$

$\sqcup(\mathsf{k}^{\delta(\mathbf{T}(\mathsf{k}))}, \mathsf{k}^{\delta(\mathbf{T}(\mathsf{k}))}) = \mathsf{k}^{\delta(\mathbf{T}(\mathsf{k}))}$

$\sqcup((\lambda x.M)^{\xi\to\phi}, (\lambda x.N)^{\nu\to\psi}) = (\lambda x.P)^{\sqcup(\xi\to\phi,\nu\to\psi)}$, where P is such that $P^{\sqcup(\phi,\psi)} = \sqcup(M^{\phi}, N^{\psi})$

$\sqcup((F\{M^{\phi_1};\cdots;M^{\phi_m}\})^{\phi}, (G\{N^{\psi_1};\cdots;N^{\psi_n}\})^{\psi}) =$
$\quad (P\ (\sqcup(\{M^{\phi_1};\cdots;M^{\phi_m}\}, \{N^{\psi_1};\cdots;N^{\psi_n}\})))^{\sqcup(\phi,\psi)},$
$\quad$ where P is such that $P^{\sqcup(\phi_1\wedge\cdots\wedge\phi_m\to\phi, \psi_1\wedge\cdots\wedge\psi_n\to\psi)} = \sqcup(F^{\phi_1\wedge\cdots\wedge\phi_m\to\phi}, G^{\psi_1\wedge\cdots\wedge\psi_n\to\psi})$

$\sqcup(\langle M_1, M_2\rangle^{\phi_1\times\phi_2}, \langle N_1, N_2\rangle^{\psi_1\times\psi_2}) = \langle P_1, P_2\rangle^{\sqcup(\phi_1,\psi_1)\times\sqcup(\phi_2,\psi_2)}$,
$\quad$ where, for $i \in \{1,2\}$, P_i is such that $P_i^{\sqcup(\phi_i,\psi_i)} = \sqcup(M_i^{\phi_i}, N_i^{\psi_i})$

$\sqcup((\mathsf{prj}_i M^{\phi_{3-i}})^{\phi_i}, (\mathsf{prj}_i N^{\psi_{3-i}})^{\psi_i}) = (\mathsf{prj}_i P^{\sqcup(\phi_{3-i},\psi_{3-i})})^{\sqcup(\phi_i,\psi_i)}$,
$\quad$ where P is such that $P^{\sqcup(\phi_1\times\phi_2,\psi_1\times\psi_2)} = \sqcup(M^{\phi_1\times\phi_2}, N^{\psi_1\times\psi_2})$

$\sqcup((\mathsf{fix}\, x.\{M^{\phi_1};\cdots;M^{\phi_m}\})^{\phi_1}, (\mathsf{fix}\, x.\{N^{\psi_1};\cdots;N^{\psi_n}\})^{\psi_1}) =$
$\quad (\mathsf{fix}\, x.\ \sqcup(\{M^{\phi_1};\cdots;M^{\phi_m}\}, \{N^{\psi_1};\cdots;N^{\psi_n}\})^{\sqcup(\phi_1,\psi_1)}$

$\sqcup((\mathsf{ifz}\, B\ \mathsf{then}\ M_1^{\phi_1}\ \mathsf{else} M_2^{\phi_2})^{\phi}, (\mathsf{ifz}\, C\ \mathsf{then}\ N_1^{\psi_1}\ \mathsf{else} N_2^{\psi_2})^{\psi}) =$
$\quad (\mathsf{ifz}\, Q\ \mathsf{then}\ \sqcup(M_1^{\phi_1}, N_1^{\psi_1})\ \mathsf{else}\ \sqcup(M_2^{\phi_2}, N_2^{\psi_2}))^{\sqcup(\phi,\psi)},$
$\quad$ where Q is such that $Q^{\delta^{\mathsf{nat}}} = \sqcup(B^{\delta^{\mathsf{nat}}}, C^{\delta^{\mathsf{nat}}})$

$\sqcup(\{M^{\phi_1};\cdots;M^{\phi_m}\}, \{N^{\psi_1};\cdots;N^{\psi_n}\}) = \{\sqcup(M^{\phi_i}, N^{\phi_j})\}_{i\in\{1,\ldots,m\}, j\in\{1,\ldots,n\}}$

Figure 14: Operation $\sqcup$ on $\Lambda^{\vdash}_{\mathbf{T}}(M)$

- The remaining cases are symmetric.

□

Lemma A.9 *For every finite set of variables Γ, for every Σ_1, Σ_2, $\Sigma_3 \in \mathbf{bases}^{\vdash}(\Gamma)$.*

1. *$\Sigma_1 \sqsubseteq \sqcup(\Sigma_1, \Sigma_2)$ and $\Sigma_2 \sqsubseteq \sqcup(\Sigma_1, \Sigma_2)$.*
2. *If $\Sigma_1 \sqsubseteq \Sigma_3$ and $\Sigma_2 \sqsubseteq \Sigma_3$, then $\sqcup(\Sigma_1, \Sigma_2) \sqsubseteq \Sigma_3$.*

Proof. Immediate from Lemma A.8. □

Lemma A.10 *For every PCFP term P and for every finite set of variables Γ such that $\Gamma \supseteq \mathrm{FV}(P)$, for every $\Sigma_1 \vdash P_1^{\chi_1}$, $\Sigma_2 \vdash P_2^{\chi_2}$, $\Sigma_3 \vdash P_3^{\chi_3} \in \mathbf{typings}^{\vdash}(P, \Gamma)$.*

1. *$\Sigma_1 \vdash P_1^{\chi_1} \sqsubseteq \sqcup(\Sigma_1, \Sigma_2) \vdash \sqcup(P_1^{\chi_1}, P_2^{\chi_2})$ and $\Sigma_2 \vdash P_2^{\chi_2} \sqsubseteq \sqcup(\Sigma_1, \Sigma_2) \vdash \sqcup(P_1^{\chi_1}, P_2^{\chi_2})$.*

2. *If* $\Sigma_2 \vdash P_2^{\chi_2} \sqsubseteq \Sigma_3 \vdash P_3^{\chi_3}$ *and* $\Sigma_2 \vdash P_2^{\chi_2} \sqsubseteq \Sigma_3 \vdash P_3^{\chi_3}$, *then* $\sqcup(\Sigma_1, \Sigma_2) \vdash \sqcup(P_1^{\chi_1}, P_2^{\chi_2}) \sqsubseteq \Sigma_3 \vdash P_3^{\chi_3}$.

Proof. Both point 1 and point 2 by structural induction on $\sqsubseteq$-derivations. We consider only point 2.

If either the derivation of $\Sigma_2 \vdash P_2^{\chi_2} \sqsubseteq \Sigma_3 \vdash P_3^{\chi_3}$ or the derivation of $\Sigma_2 \vdash P_2^{\chi_2} \sqsubseteq \Sigma_3 \vdash P_3^{\chi_3}$ ends by rule (ω) the proof is immediate. Otherwise both the derivations must end by the same rule. We consider only the case of rule (Fix). In such case, we have $P_1 = \mathsf{fix}\, x.\{M^{\phi_1}; \cdots; M^{\phi_m}\}$, $P_2 = \mathsf{fix}\, x.\{N^{\psi_1}; \cdots; N^{\psi_n}\}$, $P_3 = \mathsf{fix}\, x.\{Q^{\eta_1}; \cdots; Q^{\eta_q}\}$, $\chi_1 = \phi_1$, $\chi_2 = \psi_1$, and $\chi_3 = \eta_1$. By definition of $\sqsubseteq$, for all $h \in \{1, \ldots, q\}$,

- for some $j_1, \ldots, j_q \in \{1, \ldots, n\}$,
  $\Sigma_1, x : \phi_1 \wedge \cdots \wedge \phi_m \vdash M^{\phi_{j_h}} \sqsubseteq \Sigma_3, x : \eta_1 \wedge \cdots \wedge \eta_q \vdash Q^{\eta_h}$,
- for some $i_1, \ldots, i_q \in \{1, \ldots, m\}$,
  $\Sigma_2, x : \psi_1 \wedge \cdots \wedge \psi_n \vdash M^{\psi_{i_h}} \sqsubseteq \Sigma_3, x : \eta_1 \wedge \cdots \wedge \eta_q \vdash Q^{\eta_h}$.

So, by induction, we have that, for all $h \in \{1, \ldots, q\}$,

$$\sqcup(\Sigma_1, x : \phi_1 \wedge \cdots \wedge \phi_m \vdash M^{\phi_{j_h}}, \Sigma_2, x : \psi_1 \wedge \cdots \wedge \psi_n \vdash M^{\psi_{i_h}}) \sqsubseteq \Sigma_3, x : \eta_1 \wedge \cdots \wedge \eta_q \vdash Q^{\eta_h},$$

which, by rule (Fix), gives the result. □

For every PCFP term M of type ρ and for every finite set of variables Γ such that $\Gamma \supseteq \mathrm{FV}(M)$, consider the following sets of equivalence classes of the relation $\equiv$:

$$\begin{array}{lcl} \mathbf{L}(\rho)/\equiv & = & \{[\phi]^{\equiv} \mid \phi \in \mathbf{L}(\rho)\} \ , \\ \mathbf{bases}^{\vdash}(\Gamma)/\equiv & = & \{[\Sigma]^{\equiv} \mid \Sigma \text{ is a basis and } \epsilon(\Sigma) = \Gamma\} \ , \\ \mathbf{typings}^{\vdash}(M, \Gamma)/\equiv & = & \{[\Sigma \vdash M'^{\phi}]^{\equiv} \mid \Sigma \vdash M'^{\phi} \text{ is a } \vdash\text{-typing of } M \text{ and } \epsilon(\Sigma) = \Gamma\} \ . \end{array}$$

Let $\sqsubseteq^{\equiv}$ and $\sqcup^{\equiv}$ be the partial order relation and the operation induced by $\sqsubseteq$ and $\sqcup$, respectively, on the sets of equivalence classes $\mathbf{L}(\rho)/\equiv$, $\mathbf{bases}^{\vdash}(\Gamma)/\equiv$, and $\mathbf{typings}^{\vdash}(M, \Gamma)/\equiv$. By definition we have that $\sqsubseteq^{\equiv}$ is such that:

- $[\phi_1]^{\equiv} \sqsubseteq^{\equiv} [\phi_2]^{\equiv}$ if and only if $\phi_1 \sqsubseteq \phi_2$
- $[\Sigma_1]^{\equiv} \sqsubseteq^{\equiv} [\Sigma_2]^{\equiv}$ if and only if $\Sigma_1 \sqsubseteq \Sigma_2$
- $[\Sigma_1 \vdash M_1^{\phi_1}]^{\equiv} \sqsubseteq^{\equiv} [\Sigma_2 \vdash M_2^{\phi_2}]^{\equiv}$ if and only if $\Sigma_1 \vdash M_1^{\phi_1} \sqsubseteq \Sigma_2 \vdash M_2^{\phi_2}$

and $\sqcup^{\equiv}$ is such that:

- $\sqcup^{\equiv}([\phi_1]^{\equiv}, [\phi_2]^{\equiv}) = [\sqcup(\phi_1, \phi_2)]^{\equiv}$
- $\sqcup^{\equiv}([\Sigma_1]^{\equiv}, [\Sigma_2]^{\equiv}) = [\sqcup(\Sigma_1, \Sigma_2)]^{\equiv}$
- $\sqcup^{\equiv}([\Sigma_1 \vdash M_1^{\phi_1}]^{\equiv}, [\Sigma_2 \vdash M_2^{\phi_2}]^{\equiv}) = [(\sqcup(\Sigma_1, \Sigma_2) \vdash \sqcup(M_1^{\phi_1}, M_2^{\phi_2})]^{\equiv}$.

Lemma A.11 *For every PCFP type* $\rho \in \mathbf{T}$, *the set* $\mathbf{L}(\rho)/\equiv$ *is a finite complete lattice with partial order relation* $\sqsubseteq^{\equiv}$, *join operation* $\sqcup^{\equiv}$, *bottom element* $[\delta(\rho)]^{\equiv}$, *top element* $[\omega^{\rho}]^{\equiv}$.

Proof. First observe that, for every type ρ, if n is the number of elements in $\mathbf{L}(\rho)/\equiv$, then, for every conjunction $\phi_1 \wedge \cdots \wedge \phi_m$ of $m > n$ e-types of underlying type ρ, we have that $[\phi_1 \wedge \cdots \wedge \phi_m]^{\equiv} = [\phi_{i_1} \wedge \cdots \wedge \phi_{i_n}]^{\equiv}$, for some $i_1, \ldots, i_n \in \{1, \ldots, n\}$. We can now prove, by structural induction on types, that $\mathbf{L}(\rho)/\equiv$ is finite.

$\rho = \mathsf{nat}$**.** Immediate, since there are only two classes: $\mathbf{L}(\mathsf{nat})/\equiv = \{[\delta^{\mathsf{nat}}]^{\equiv}, [\omega^{\mathsf{nat}}]^{\equiv}\}$.

$\rho = \sigma \to \tau$**.** For all the conjunctions $\xi_1^\sigma, \xi_2^\sigma$ and for every ϕ_1^τ, ϕ_2^τ, we have that $[\xi_1^\sigma]^{\equiv} = [\xi_2^\sigma]^{\equiv}$ and $[\phi_1^\tau]^{\equiv} = [\phi_2^\tau]^{\equiv}$ imply $[\xi_1^\sigma \to \phi_1^\tau]^{\equiv} = [\xi_2^\sigma \to \phi_2^\tau]^{\equiv}$. So the result follows by induction.

$\rho = \rho_1 \times \rho_2$**.** Similar.

Now the result follows directly by Lemma A.8. □

Lemma A.12 *For every finite set of variable Γ, the set $\mathbf{bases}^{\vdash}(\Gamma)/\equiv$ is a finite complete lattice with partial order relation $\sqsubseteq^{\equiv}$, join operation $\sqcup^{\equiv}$, bottom element $[\delta(\Gamma)]^{\equiv}$, and top element $[\omega(\Gamma)]^{\equiv}$.*

Proof. Immediate from Lemma A.11. □

Lemma A.13 *For every PCFP term P of type ρ, for every finite set of variable Γ such that $\Gamma \supseteq \mathrm{FV}(P)$, the set $\mathbf{typings}^{\vdash}(P, \Gamma)/\equiv$ is a finite complete lattice with partial order relation $\sqsubseteq^{\equiv}$, join operation $\sqcup^{\equiv}$, bottom element $[\delta(\Gamma) \vdash \delta(P)]^{\equiv}$, and top element $[\omega(\Gamma) \vdash P^{\omega^\rho}]^{\equiv}$.*

Proof. First observe that, for every PCFP term P of type ρ and for every finite set of variable Γ such that $\Gamma \supseteq \mathrm{FV}(P)$, if n is the number of elements in $\mathbf{typings}^{\vdash}(P, \Gamma)/\equiv$, then, for every sequence $\{{P_1}^{\phi_1}; \cdots; {P_1}^{\phi_m}\}$ such that, for some $\Sigma \in \mathbf{bases}^{\vdash}(\Gamma)$ and for all $l \in \{1, \ldots, n\}$, $\Sigma \vdash {P_l}^{\phi_l}$, we have that $m > n$ implies

$$[\{{P_1}^{\phi_1}; \cdots; {P_1}^{\phi_m}\}]^{\equiv} \;=\; [\{{P_{i_1}}^{\phi_{i_1}}; \cdots; {P_{i_n}}^{\phi_{i_n}}\}]^{\equiv},$$

for some $i_1, \ldots, i_n \in \{1, \ldots, n\}$. We can now prove, by structural induction on terms, that the set $\mathbf{typings}^{\vdash}(P, \Gamma)/\equiv$ is finite. We consider only three cases.

$P = x^\rho$. Any derivation of a $\vdash$-typing $\Sigma \vdash P^\psi$ must end with an application of either rule (ω) or rule (Var). In both cases we have a finite number of $\equiv$-classes. In the former case by Lemma A.12, in the latter by Lemmas A.12 and A.11.

$P = MN$. Any derivation of a $\vdash$-typing $\Sigma \vdash P^\psi$ must end by either rule (ω) or rule ($\to$ E). In both cases we have a finite number of $\equiv$-classes. The former case is by Lemma A.12. The latter is straightforward by induction, since, for any $\Sigma \vdash M^{\phi_1 \wedge \cdots \wedge \phi_m \to \psi}$, $\Sigma \vdash N^{\phi_i}$ $(1 \le i \le m)$, $\Sigma' \vdash {M'}^{\phi'_1 \wedge \cdots \wedge \phi'_n \to \psi'}$, and $\Sigma' \vdash {N'}^{\phi'_j}$ $(1 \le j \le n)$: $[\Sigma \vdash M^{\phi_1 \wedge \cdots \wedge \phi_m \to \psi}]^{\equiv} = [\Sigma' \vdash {M'}^{\phi'_1 \wedge \cdots \wedge \phi'_n \to \psi'}]^{\equiv}$ and $[\{N^{\phi_1}; \cdots; N^{\phi_m}\}]^{\equiv} = [\{{N'}^{\phi'_1}; \cdots; {N'}^{\phi'_n}\}]^{\equiv}$ imply $[\Sigma \vdash (M\{N^{\phi_1}; \cdots; N^{\phi_n}\})^{\psi}]^{\equiv} = [\Sigma' \vdash (N'\{{N'}^{\phi'_1}; \cdots; {N'}^{\phi'_n}\})^{\psi'}]^{\equiv}$.

$P = \mathsf{fix}\, x.M$. Any derivation of a $\vdash$-typing $\Sigma \vdash P^{\phi_1}$ must end by either rule (ω) or rule (Fix). In both cases we have a finite number of $\equiv$-classes. The former case is by Lemma A.12. The latter is straightforward by induction, since, for any $\Sigma \vdash M^{\phi_i}$ $(1 \leq i \leq m)$ and $\Sigma' \vdash M'^{\phi'_j}$ $(1 \leq j \leq n)$, we have that: $[\phi_1]^{\equiv} = [\phi'_1]^{\equiv}$ and $[\{M^{\phi_1}; \cdots; M^{\phi_m}\}]^{\equiv} = [\{M'^{\phi'_1}; \cdots; M'^{\phi'_n}\}]^{\equiv}$ imply $[\Sigma \vdash (\mathsf{fix}\, x.\{M^{\phi_1}; \cdots; M^{\phi_n}\})^{\phi_1}]^{\equiv} = [\Sigma' \vdash (\mathsf{fix}\, x.\{M'^{\phi'_1}; \cdots; M'^{\phi'_n}\})^{\phi'_1}]^{\equiv}$.

Now the result follows directly by Lemma A.10. □

Restatement of Theorem 5.19. *For every PCFP term M, the set* $\mathbf{faithful}^{\vdash}(M)/\equiv$ *is a finite complete lattice with partial order relation $\sqsubseteq^{\equiv}$ and bottom element $[\delta(\mathrm{FV}(M)) \vdash \delta(M)]^{\equiv}$.*

Proof. By Lemma A.13, since $\mathbf{faithful}^{\vdash}(M)/\equiv$ is a sublattice of $\mathbf{typings}^{\vdash}(M, \mathrm{FV}(M))/\equiv$. □

Restatement of Theorem 5.21 *For every PCFP term M, $\mathbf{O}^{\vdash}(M) = \mathbf{O}^{\vdash}(\mathbf{O}^{\vdash}(M))$.*

Proof. We will show that, if $\Sigma \vdash M'$ is an optimal faithful $\vdash$-typing of M and $N = \mathbf{O}^{\vdash}(M)$, then $\Sigma' \vdash N'$ (where N' is obtained from M' by replacing by d the subterms which are replaced by d in N, and $\Sigma' = \{x : \xi \in \Sigma \mid x \in \mathrm{FNDV}(N)\} \cup \{\mathsf{d}^{\rho} : \omega^{\rho} \mid \mathsf{d}^{\rho} \in \mathrm{DV}(N)\}$) is an optimal faithful $\vdash$-typing of N.

From Lemma 5.13.1 we have that $\Sigma' \vdash N'$ is a faithful $\vdash$-typing of N. Assume, by contraposition, that it is not optimal. This implies that an optimal faithful $\vdash$-typing of N, say $\Sigma'' \vdash N''$ is such that an ω-e-type is always assigned to some subexpression of N different from d. Then it is possible to define (starting from the derivation $\Sigma'' \vdash N''$) a faithful $\vdash$-typing of M, $\Sigma''' \vdash M''$, with

$$\Sigma''' = \{x : \xi \in \Sigma'' \mid x \in \mathrm{FV}(M)\} \cup \{x^{\rho} : \omega^{\rho} \mid x^{\rho} \in \mathrm{FV}(M) - \mathrm{FV}(N)\},$$

such that $\Sigma''' \vdash M'' \not\sqsubseteq \Sigma \vdash M'$. This contradicts the hypothesis that $\Sigma \vdash M'$ is optimal. □