IO vs OI in Higher-Order Recursion Schemes

Axel Haddad

LIGM (Université Paris Est & CNRS),
LIAFA (Université Paris Diderot & CNRS)

Abstract We propose a study on the modes of derivation of higher-order recursion schemes, proving that value trees obtained from schemes using innermost-outermost derivations are the same as those obtained using unrestricted derivations.

1 Introduction

Recursion schemes have been first considered as a model of computation, representing the syntactical aspects of a recursive program [15,2,3,4] . At first, (order-1) schemes used to model simple recursive programs whose functions only take values as input (and not functions). Since, higher-order versions of recursion schemes [11,5,6,7,8,9] have been studied.

More recently, recursion schemes were studied as generators of infinite ranked trees and the focus is then to decide logical properties of those infinite trees [8,1,10,13,14].

As for programming languages, the question of evaluation plays a central role. Indeed, different policies results in different evaluation trees [8,9,7]. There are two main evaluation policies for schemes: outermost-innermost derivations (OI) and inner-outermost derivations (IO), respectively corresponding to call-by-need and call-by-value in programming languages.

The standardization theorem for the lambda-calculus implies that for any scheme, outermost-innermost derivations (OI) lead to the same tree as unrestricted derivation. However, this is not the case for IO derivations. In this paper we establish that the trees produced using schemes with IO policy are the same as those produced using schemes with OI policy. For a given scheme of order n, we can use a lazying transformation, to construct a new scheme of order $n+1$ in which IO derivations will be the same as OI derivations in the initial scheme (Section 3). This increase of the order is shown to be unavoidable for order 1 schemes and conjectured to be so at any order.

Conversely, in order to turn a scheme into another one in which unrestricted derivations lead to the same tree as IO derivations in the initial scheme, we adapt Kobayashi's recent results on higher order recursion scheme (HORS) model-checking [13], to compute some key properties over terms (Section 4.1). Then we embed these properties into the scheme while preserving the order obtaining a scheme in which OI and IO derivations produce the same tree (Section 4.2).

Finally, we show that this technique can be used to improve efficiency of derivation, for example by preventing non-productive derivations, i.e. derivations that would not produce anything in the value tree.

2 Preliminaries

Given a set A we write A^* the set of words of A. We write uv the concatenation of the words u and v, and we say that u is a prefix of w if there exists v such that $uv = w$. We say that a subset S of A^* is prefix-closed if for all $w \in S$, S contains all prefixes of w.

Types are defined by the grammar $\tau ::= o \mid \tau \to \tau$ and o is called the ***ground type***. Considering that $\to$ is associative to the right i.e. $\tau_1 \to (\tau_2 \to \tau_3)$ can be written $\tau_1 \to \tau_2 \to \tau_3$, any type τ can be written uniquely as $\tau_1 \to ... \to \tau_k \to o$. The integer k is called the ***arity*** of τ. We inductively define the ***order of a type*** by $\mathrm{order}(o) = 0$ and $\mathrm{order}(\tau_1 \to \tau_2) = \max(\mathrm{order}(\tau_1) + 1, \mathrm{order}(\tau_2))$. For instance $o \to o \to o \to o$ is an order 1 type of arity 3, $(o \to o) \to (o \to o)$, that can also be written $(o \to o) \to o \to o$ is a type of order 2 and arity 2. We use the notation $\tau^\ell \to \tau'$ as a shorthand for $\underbrace{\tau \to ... \to \tau}_{\ell\ times} \to \tau'$.

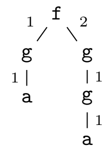
Let Γ be a finite set of typed symbols, i.e. to each symbol is associated a type, and we let Γ^τ denotes the set of symbols of type τ. For all type τ, we define the set $\mathcal{T}^\tau(\Gamma)$ of ***terms*** of type τ as the smallest set satisfying: $\Gamma^\tau \subseteq \mathcal{T}^\tau(\Gamma)$ and $\bigcup_{\tau'}\{t\ s \mid t \in \mathcal{T}^{\tau' \to \tau}(\Gamma), s \in \mathcal{T}^{\tau'}(\Gamma)\} \subseteq \mathcal{T}^\tau(\Gamma)$. We write $\mathcal{T}(\Gamma)$ for the set of terms of any type, and $t : \tau$ if t has type τ. The arity of a term t, $arity(t)$, is the arity of its type. Remark that any term t can be uniquely written as $t = \alpha\ t_1...t_k$ with $\alpha \in \Gamma$ and t_1, ..., t_k some terms. We say that α is the ***head*** of the term t.

Example 1. Let $\Gamma = \{F : (o \to o) \to o \to o\ ,\ G : o \to o \to o\ ,\ H : (o \to o)\ ,\ \mathtt{a} : o\}$: $F\ H$ and $G\ \mathtt{a}$ are terms of type $o \to o$; $F(G\ \mathtt{a})\ (H\ (H\ \mathtt{a}))$ is a term of type o; $F\ \mathtt{a}$ is not a term since F is expecting a first argument of type $o \to o$ while $\mathtt{a}$ is of type o.

Let $t : \tau$, $t' : \tau'$ be two terms, $x : \tau'$ be a symbol in $\Gamma^{\tau'}$. We write $t_{[x \mapsto t']} : \tau$ the term obtained by substituting all occurrences of x by t' in the term t, it is defined by induction: $x_{[x \mapsto t']} = t'$, $y_{[x \mapsto t']} = y$ for $y \in \Gamma$ and $y \neq x$, and $t_1\ t_{2[x \mapsto t']} = t_{1[x \mapsto t']}\ t_{2[x \mapsto t']}$. A ***$\tau$-context*** is a term $C[\bullet^\tau] \in \mathcal{T}(\Gamma \uplus \{\bullet^\tau : \tau\})$ containing exactly one occurrence of $\bullet^\tau$; it can be seen as an application turning a term into another, such that for all $t : \tau$, $C[t] = C[\bullet^\tau]_{[\bullet^\tau \mapsto t]}$. In general we will only consider ground type context where $\tau = o$ and we will omit to specify the type when it is clear. For instance, if $C[\bullet] = F\ \bullet\ \ (H\ (H\ \mathtt{a}))$ and $t' = G\ \mathtt{a}$ then $C[t'] = F\ (G\ \mathtt{a})\ (H\ (H\ \mathtt{a}))$.

Let Σ be a set of symbols of order at most 1 (i.e. each symbols has type o or $o \to ... \to o$) and $\perp : o$ be a fresh symbol. A ***tree*** t over $\Sigma \uplus \{\perp\}$ is a mapping $t : dom^t \to \Sigma \uplus \{\perp\}$, where dom^t is a prefix-closed subset of $\{1, ..., m\}^*$, m being the maximum arity of Σ, such that if $u \in dom^t$ and $t(u) = a$ then $\{j \mid uj \in dom^t\} = \{1, ..., arity(a)\}$. Note that there is a direct bijection between ground terms of $\mathcal{T}^o(\Sigma \uplus \{\perp\})$ and finite trees . Hence we may later see ground terms over $\Sigma \uplus \{\perp\}$ as trees.

Example 2. The term `f (g a) (g (g a))` with $\mathtt{f} : o^2 \to o$, $\mathtt{g} : o \to o$ and $\mathtt{a} : o$ is seen as the tree on the right.



We define the prefix partial order $\sqsubseteq$ over trees as the smallest relation satisfying $\bot \sqsubseteq t$ and $t \sqsubseteq t$ for any tree t, and $a\ t_1 ... t_k \sqsubseteq a\ t'_1 ... t'_k$ iff $t_i \sqsubseteq t'_i$. Given a (possibly infinite) sequence of trees $t_0, t_1, t_2, ...$ such that $t_i \sqsubseteq t_{i+1}$ for all i, one can prove that the set of all t_i has a supremum that is called the ***limit tree*** of the sequence.

A ***higher order recursion scheme (HORS)*** $\mathcal{G} = \langle \mathcal{V}, \Sigma, \mathcal{N}, \mathcal{R}, S \rangle$ is a tuple such that: $\mathcal{V}$ is a finite set of typed symbols called ***variables***; Σ is a finite set of typed symbols of order at most 1, called the ***set of terminals***; $\mathcal{N}$ is a finite set of typed symbols called the ***set of non-terminals***; $\mathcal{R}$ is a set of ***rewrite rules***, one per non-terminal $F : \tau_1 \to ... \to \tau_k \to o \in \mathcal{N}$, of the form $F\ x_1\ ...\ x_k\ \to\ e$ with $e : o \in \mathcal{T}(\Sigma \uplus \mathcal{N} \uplus \{x_1, ..., x_k\})$; $S : o \in \mathcal{N}$ is the ***initial non-terminal***.

A term r is called a ***redex*** if $r : o$ and $r = F\ t_1\ ...\ t_k$ with $F \in \mathcal{N}$. If $F\ x_1 ... x_k \to e \in \mathcal{R}$, we say that the redex r can be rewritten in $e_{[x_1 \mapsto t_1] ... [x_k \mapsto t_k]}$, and that a term t is rewritten in a term t' if t contains an occurrence of a redex r and t' is the same as t where the occurrence of r is rewritten in $e_{[x_1 \mapsto t_1] ... [x_k \mapsto t_k]}$. Formally, we define the ***rewriting relation*** $\to_\mathcal{G}\ \in\ \mathcal{T}(\Sigma \uplus \mathcal{N})^2$ (or just $\to$ when $\mathcal{G}$ is clear) as $t \to_\mathcal{G} t'$ if and only if there exists a context $C[\bullet]$, a rewrite rule $F\ x_1 ... x_k \to e$, and a redex $F\ t_1\ ...\ t_k : o$ such that $t = C[F\ t_1 ... t_k]$ and $t' = C[e_{[x_1 \mapsto t_1] ... [x_k \mapsto t_k]}]$. Finally we define a ***derivation***, as a possibly infinite sequence of terms linked by the rewriting relation $t_0 \to_\mathcal{G} t_1 \to_\mathcal{G} t_2 \to_\mathcal{G} ...$, and $\to_\mathcal{G}^*$ to be the reflexive and transitive closure of $\to_\mathcal{G}$. We say that a derivation is maximal if it is etiher infinite or it is finite and the last term cannot be rewritten.

We define inductively the ***$\bot$-transformation*** $(\cdot)^\bot\ :\ \mathcal{T}^o(\mathcal{N} \uplus \Sigma) \to \mathcal{T}^o(\Sigma \uplus \{\bot : o\})$, used to transform a ground typed term into a finite tree, as for all $F \in \mathcal{N}$ $(F\ t_1\ ...\ t_k)^\bot\ =\ \bot$ and for all $a \in \Sigma$ $(a\ t_1\ ...\ t_k)^\bot\ =\ a\ t_1^\bot\ ...\ t_k^\bot$. Let $t_0 = S \to_G t_1 \to_\mathcal{G} t_2 \to_\mathcal{G} ...$ be a derivation, then one can check that $(t_0)^\bot \sqsubseteq (t_1)^\bot \sqsubseteq (t_2)^\bot \sqsubseteq ...$, hence it admits a limit. One can prove that the set of all such limit trees has a greatest element that we denote $\|\mathcal{G}\|$ and refer to as the ***value tree*** of $\mathcal{G}$. Note that $\|\mathcal{G}\|$ is the supremum of $\{t^\bot \mid S \to^* t\}$. Given a term $t : o$, we denote by $\mathcal{G}_t$ the scheme obtained by transforming $\mathcal{G}$ such that it starts derivations with the term t, formally, $\mathcal{G}_t = \langle \mathcal{V}, \Sigma, \mathcal{N} \uplus \{S'\}, \mathcal{R} \uplus \{S' \to t\}, S' \rangle$. Remark that if $t \to t'$ then $\|\mathcal{G}_t\| = \|\mathcal{G}_{t'}\|$.

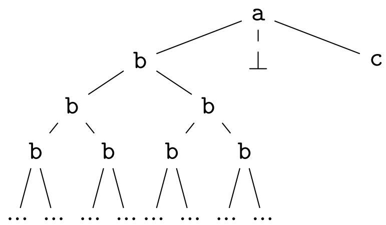
Example 3. Let $\mathcal{G} = \langle \mathcal{V}, \Sigma, \mathcal{N}, \mathcal{R}, S \rangle$ be the scheme such that: $\mathcal{V} = \{x : o, \phi : o \to o, \psi : (o \to o) \to o \to o\}$, $\Sigma = \{\mathtt{a} : o^3 \to o, \mathtt{b} : o \to o \to o, \mathtt{c} : o\}$. The set of non-terminals is $\mathcal{N} = \{F : \big((o \to o) \to o \to o\big) \to (o \to o) \to o \to o, H : (o \to o) \to o \to o, I, J, K : o \to o, S : o\}$, and $\mathcal{R}$ contains the following rewrite rules:

$$
\begin{array}{llllll}
F\ \psi\ \phi\ x & \to \psi\ \phi\ x & I\ x & \to x & H\ \phi\ x & \to \mathtt{a}\ (J\ x)\ (K\ x)\ (\phi\ x) \\
J\ x & \to \mathtt{b}\ (J\ x)\ (J\ x) & K\ x & \to K\ (K\ x) & S & \to F\ H\ I\ \mathtt{c}
\end{array}
$$

Here is an example of finite derivation:

$$\begin{array}{ccccccc} S & \to & F\ H\ I\ \mathtt{c} & \to & H\ I\ \mathtt{c} & \to & \mathtt{a}\ (J\ \mathtt{c})\ (K\ \mathtt{c})\ (I\ \mathtt{c}) \\ & & \to \quad \mathtt{a}\ (J\ \mathtt{c})\ (K\ (K\ \mathtt{c}))\ (I\ \mathtt{c}) & & \to & & \mathtt{a}\ (J\ \mathtt{c})\ (K\ (K\ (K\ \mathtt{c})))\ (I\ \mathtt{c}) \end{array}$$

If one extends it by always rewriting a redex of head K, its limit is the tree $\mathtt{a}\ \bot\ \bot\ \bot$, but this tree is not the value tree of $\mathcal{G}$. The value tree $\|\mathcal{G}\|$ is depicted on the right.



Evaluation Policies

We now put constraints on the derivations we allow. If there are no constraints, then we say that the derivations are unrestricted and we let $\mathrm{Acc}^{\mathcal{G}} = \{t : o \mid S \to^* t\}$ be the set of accessible terms using unrestricted derivations. Intuitively an outermost-innermost rewriting is a rewriting where the redex chosen to be rewritten has no redex above it in the term (it is an outermost redex), in an innermost-outermost rewriting the redex chosen has no redex as a subterm (it is an innermost redex).

Formally given a term $t \to t'$ such that $t = C[F\ s_1\ ...\ s_k]$ and $t' = C[e_{[\forall j\ x_j \mapsto s_j]}]$ with $F\ x_1\ ...\ x_k \to e \in \mathcal{R}$:

– We say that $t \to t'$ is an ***outermost-innermost*** (OI) rewriting (written $t \to_{\mathrm{OI}} t'$) if there is no redex containing the occurrence of $\bullet$ as a subterm of $C[\bullet]$.
– We say that $t \to t'$ is an ***innermost-outermost*** (IO) rewriting (written $t \to_{\mathrm{IO}} t'$), if for all $j \in \{1, ..., k\}$ there is no redex as a subterm of s_j.

Let $\mathrm{Acc}^{\mathcal{G}}_{\mathrm{OI}} = \{t : o \mid S \to^*_{\mathrm{OI}} t\}$ be the set of accessible terms using OI derivations and $\mathrm{Acc}^{\mathcal{G}}_{\mathrm{IO}} = \{t : o \mid S \to^*_{\mathrm{IO}} t\}$ be the set of accessible terms using IO derivations. There exists a supremum of $\mathrm{Acc}^{\mathcal{G}}_{\mathrm{OI}}$ (resp. $\mathrm{Acc}^{\mathcal{G}}_{\mathrm{IO}}$) which is the maximum of the limit trees of OI derivations(resp. IO derivations). We write it $\|\mathcal{G}\|_{\mathrm{OI}}$ (resp. $\|\mathcal{G}\|_{\mathrm{IO}}$). For all recursive scheme $\mathcal{G}$, $(\mathrm{Acc}^{\mathcal{G}})^{\bot} = (\mathrm{Acc}^{\mathcal{G}}_{\mathrm{OI}})^{\bot}$, in particular $\|\mathcal{G}\|_{\mathrm{OI}} = \|\mathcal{G}\|$. But $\|\mathcal{G}\|_{\mathrm{IO}} \sqsubseteq \|\mathcal{G}\|$ and in general, the equality does not hold (see Example 4).

3 From OI to IO

Fix a recursion scheme $\mathcal{G} = \langle \mathcal{V}, \Sigma, \mathcal{N}, \mathcal{R}, S \rangle$. Our goal is to define another scheme $\overline{\mathcal{G}} = \langle \overline{\mathcal{V}}, \Sigma \uplus \{\mathtt{k} : o\}, \overline{\mathcal{N}}, \overline{\mathcal{R}}, I \rangle$ such that $\|\overline{\mathcal{G}}\|_{\mathrm{IO}} = \|\mathcal{G}\|$. The idea is to add an extra argument $(\mathtt{k} : o)$ to each non-terminal, that will be required to rewrite it (hence the types are changed and the order is augmented by 1). We feed this argument to the outermost non-terminal, and duplicate it to subterms only if the head of the term is a terminal. Hence all derivations will be OI-derivations. This is a standard transformation.

Example 4. Let $\mathcal{G} = \langle \mathcal{V}, \Sigma, \mathcal{N}, \mathcal{R}, S\rangle$ be the order-1 recursion scheme with $\Sigma = \{\mathtt{a}, \mathtt{c} : o\}$, $\mathcal{N} = \{S : o, F : o \to o \to o, H : o \to o\}$, $\mathcal{V} = \{x, y : o\}$, and the following rewrite rules:

$$S \to F\ (H\ \mathtt{a})\ \mathtt{c} \qquad F\ x\ y \to y \qquad H\ x \to H\ (H\ x)$$

Then we have $\|\mathcal{G}\|_{\text{OI}} = \mathtt{c}$ while $\|\mathcal{G}\|_{\text{IO}} = \bot$ (indeed, the only IO derivation is the following $S \to F\ (Ha)\ \mathtt{c} \to F\ (H\ (H\ a))\ \mathtt{c} \to F\ (H\ (H\ (H\ a)))\ \mathtt{c} \to ...$). The order-2 recursion scheme $\overline{\mathcal{G}} = \langle \overline{\mathcal{V}}, \Sigma \uplus \{\mathtt{k} : o\}, \overline{\mathcal{N}}, \overline{\mathcal{R}}, I\rangle$ is given by $\overline{\mathcal{N}} = \{I : o, \overline{S}, \overline{\mathtt{a}}, \overline{\mathtt{c}} : o \to o, \overline{F} : (o \to o) \to (o \to o) \to o \to o, \overline{H} : (o \to o) \to o \to o\}$,$\overline{\mathcal{V}} = \{\ell : o, \overline{x}, \overline{y} : o \to o\}$ and the following rewrite rules:

$$\begin{array}{lll} I \quad\ \ \to \overline{S}\ \mathtt{k} & \overline{S}\ \ell \to \overline{F}\ (\overline{H}\ \overline{\mathtt{a}})\ \overline{\mathtt{c}}\ \mathtt{k} & \overline{F}\ \overline{x}\ \overline{y}\ \ell \to \overline{y}\ \mathtt{k} \\ \overline{H}\ \overline{x}\ \ell \to \overline{H}\ (\overline{H}\ \overline{x})\ \mathtt{k} & \overline{\mathtt{c}}\ \ell \to \mathtt{c} & \overline{\mathtt{a}}\ \ell \quad\ \to \mathtt{a} \end{array}$$

Note that in the term $\overline{F}\ (\overline{H}\ \overline{\mathtt{a}})\ \overline{\mathtt{c}}\ \mathtt{k}$, the subterm $\overline{H}\ \overline{\mathtt{a}}$ is no longer a redex since it lacks its last argument, hence it cannot be rewritten, then the only IO derivation, which is the only unrestricted derivation is $I \to \overline{S}\ \mathtt{k} \to \overline{F}\ (\overline{H}\ \overline{\mathtt{a}})\ \overline{\mathtt{c}}\ \mathtt{k} \to \overline{\mathtt{c}}\ \mathtt{k} \to \mathtt{c}$. Therefore $\|\overline{\mathcal{G}}\|_{\text{IO}} = \|\overline{\mathcal{G}}\|_{\text{OI}} = \mathtt{c} = \|\mathcal{G}\|_{\text{OI}}$.

Formally, we define the $\overline{(\cdot)}$ ***transformation*** over types by $\overline{o} = o \to o$, and $\overline{\tau_1 \to \tau_2} = \overline{\tau_1} \to \overline{\tau_2}$. In particular, if $\tau = \tau_1 \to ... \to \tau_k \to o$ then $\overline{\tau} = \overline{\tau_1} \to ... \to \overline{\tau_k} \to o \to o$. Note that for all τ, $\text{order}(\overline{\tau}) = \text{order}(\tau) + 1$ and $\text{arity}(\overline{\tau}) = \text{arity}(\tau) + 1$.

For all $x : \tau \in \mathcal{V}$ we take $\overline{x} : \overline{\tau}$ to be a fresh variable. Let $ar_{\max}$ be the maximum arity of terminals, we take $\eta_1, ..., \eta_{ar_{\max}} : o \to o$ and $\ell : o$ to be fresh variables, and we let $\overline{\mathcal{V}} = \{\overline{x} : \overline{\tau} \mid x \in \mathcal{V}\} \uplus \{\eta_1, ..., \eta_{ar_{\max}}\} \uplus \{\ell : o\}$. Note that ℓ is the only variable of type o. For all $a : \tau \in \Sigma$ define $\overline{a} : \overline{\tau}$ as a fresh **non-terminal** for all $F : \tau \in \mathcal{N}$ define $\overline{F} : \overline{\tau}$ as a fresh non-terminal. Let $\overline{\mathcal{N}} = \{\overline{a} : \overline{\tau} \mid a \in \Sigma\} \uplus \{\overline{F} : \overline{\tau} \mid F \in \mathcal{N}\} \uplus \{I : o\}$. Note that I is the only symbol in $\overline{\mathcal{N}}$ of type o.

Let $t : \tau \in \mathcal{T}(\mathcal{V} \uplus \Sigma \uplus \mathcal{N})$, we define inductively the term $\overline{t} : \overline{\tau} \in \mathcal{T}(\overline{\mathcal{V}} \uplus \overline{\mathcal{N}})$: If $t = x \in \mathcal{V}$ (resp. $t = a \in \Sigma$, $t = F \in \mathcal{N}$), we let $\overline{t} = \overline{x} \in \overline{\mathcal{V}}$ (resp. $\overline{t} = \overline{a} \in \overline{\Sigma}$, $\overline{t} = \overline{F} \in \mathcal{N}$), if $t = t_1\ t_2 : \tau$ then $\overline{t} = \overline{t_1}\ \overline{t_2}$.

Let $F\ x_1\ ...\ x_k \to e$ be a rewrite rule of $\mathcal{R}$. We define the (valid) rule $\overline{F}\ \overline{x_1}\ ...\ \overline{x_k}\ \ell \to \overline{e}\ \mathtt{k}$ in $\overline{\mathcal{R}}$. Let $a \in \Sigma$ of arity k, we define the rule $\overline{a}\ \eta_1\ ...\ \eta_k\ \ell \to a\ (\eta_1\ \mathtt{k})\ ...\ (\eta_k\ \mathtt{k})$ in $\overline{\mathcal{R}}$. We also add the rule $I \to \overline{S}\ \mathtt{k}$ to $\overline{\mathcal{R}}$. Finally let $\overline{G} = \langle \overline{\mathcal{V}}, \Sigma \uplus \{\mathtt{k} : o\}, \overline{\mathcal{N}}, \overline{\mathcal{R}}, I\rangle$.

Lemma 1. *Any derivation of $\overline{\mathcal{G}}$ is in fact an* OI ***and*** *an* IO *derivation. Hence* $\|\overline{\mathcal{G}}\|_{\text{IO}} = \|\overline{\mathcal{G}}\|$.

Proof (Sketch). The main idea is that (1) the only redexes will be the terms that have an occurrence of $\mathtt{k}$ as last argument of the head non-terminal, which we can prove by a study on types of the terms produced during a derivation. The scheme is constructed so that $\mathtt{k}$ remains only on the outermost non-terminals, that is why any derivation is an OI derivation. Furthermore, we have that if

$t = \overline{F}\ t_1\ ...\ t_k$ `k` is a redex, then none of the t_i contain an occurence of `k`, therefore they do not contain any redex, hence (1) states that t is an innermost redex. □

Theorem 1 (OI vs IO). *Let $\mathcal{G}$ be an order-n scheme. Then one can construct an order-$(n+1)$ scheme $\overline{\mathcal{G}}$ such that $\|\mathcal{G}\| = \|\overline{\mathcal{G}}\|_{\text{IO}}$.*

Proof (Sketch). Taking into account Lemma 1, we just have to show that OI derivations in $\overline{\mathcal{G}}$ acts like OI derivations in $\mathcal{G}$, hence $\|\mathcal{G}\| = \|\overline{\mathcal{G}}\|$. Therefore, we construct some maximal derivations in $\mathcal{G}$ and in $\overline{\mathcal{G}}$, and we show that we can get one from the other, and that they lead to the same tree. □

In Example 4 the value tree of the original scheme, was a finite tree `c`, it could have been done with an order 0 scheme with one non-terminal: $S \to$ `c`. It is actually easy to show that the set of trees produced by order 0 scheme is the set of all regular trees. We show in Example 5 an order 1 scheme whose OI value tree cannot be produced by any order 1 scheme using only IO derivations.

Example 5. Let $\mathcal{G}_0 = \langle \Sigma_0, \mathcal{N}_0, \mathcal{V}_0, \mathcal{R}_0, S_0 \rangle$ be the order 1 scheme such that $\Sigma = \{\mathtt{a} : o \to o, \mathtt{b} : o^2 \to o\}$, $\mathcal{N}\{S_0, B : o, F : o \to o\}$, $\mathcal{V} = \{x : o\}$, and $\mathcal{R}_0$ contains the following rewrite rules:

$$S_0 \to F\ B \qquad F\ x \to \mathtt{b}\ x\ (F\ (\mathtt{a}\ B)) \qquad B \to \mathtt{b}\ B\ B$$

The tree $\|\mathcal{G}_0\|$ is depicted in Figure 1. The tree is not regular, then it cannot be generated by an order 0 scheme. We can show the following theorem.

Theorem 2. *There exists no order 1 scheme $\mathcal{G}$ such that $\|\mathcal{G}\|_{\text{IO}} = \|\mathcal{G}_0\|$.*

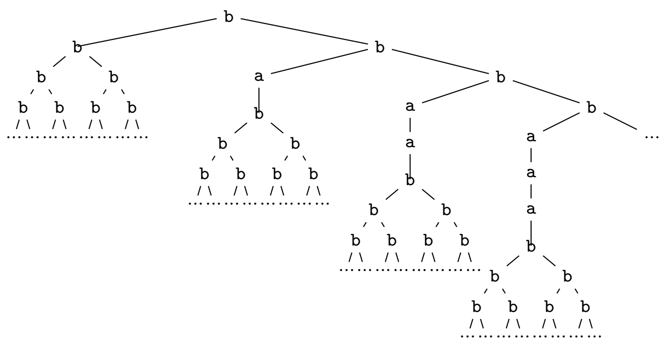


Figure 1. The tree $\|\mathcal{G}_0\|$

4 From IO to OI

The goal of this section is to transform the scheme $\mathcal{G}$ into a scheme $\mathcal{G}''$ such that $\|\mathcal{G}''\| = \|\mathcal{G}\|_{\text{IO}}$. The main difference between IO and OI derivations is that some redex would lead to $\perp$ in IO derivation while OI derivations could be more productive. For example take $F : o \to o$ such that $F\ x \to \mathtt{c}$, and $H : o$ such that $H \to \mathtt{a}\ H$, with $\mathtt{a} : o \to o$ and $c : o$ being terminal symbols. The term $F\ H$ has a unique OI derivation, $F\ H \to_{\text{OI}} \mathtt{c}$, it is finite and it leads to the value tree associated. On the other hand, the (unique) IO derivation is the following $F\ H \to F(\mathtt{a}\ H) \to F\ (\mathtt{a}\ (\mathtt{a}\ H)) \to ...$ which leads to the tree $\perp$. In general, take a redex $F\ t_1...t_k$, if for some t_i, all the IO derivations from t_i are infinite, then the IO value tree associated to the term $F\ t_1...t_k$ will be $\perp$.

The idea of the transformation is to compute an anotation (based on a type system) that helps decide if a redex would produce $\perp$ with IO derivations (Section 4.1); then we embed it into $\mathcal{G}$ and force any such redex to produce $\perp$ even with unrestricted derivations (Section 4.2).

4.1 The Type System

Given a term $t : \tau \in \mathcal{T}(\Sigma \uplus \mathcal{N})$, we define the following properties on t: $\mathcal{P}_\perp(t)$ ="The term t has type o and its associated IO valuation tree is $\perp$", and $\mathcal{P}_\infty(t)$ ="the term t has not necessarily ground type, it contains a redex r such that any maximal IO derivation from r producing it's IO valuation tree is infinite" (recall that a maximal derivation is either infinite, or finite and the last term cannot be rewritten). Note that $\mathcal{P}_\infty(t)$ is equivalent to "the term t contains a redex r such that $\|\mathcal{G}_r\|_{\text{IO}}$ is either infinite or contains $\perp$". In this section we describe a type system, inspired from the work of Kobayashi [13], that characterises the terms verifying these properties.

Let Q be the set $\{q_\perp, q_\infty\}$. Given a type τ, we define inductively the sets $(\tau)^{atom}$ and $(\tau)^\wedge$ called respectively set of ***atomic mappings*** and set of ***conjunctive mappings***: $(o)^{atom} = Q$, for all $\tau_1 \to \tau_2$, $(\tau_1 \to \tau_2)^{atom} = \{q_\infty\} \uplus \{(\tau_1)^\wedge \to (\tau_2)^{atom}\}$ and for all τ, $(\tau_1 \to \tau_2)^\wedge = \{\bigwedge\{\theta_1, ..., \theta_i\} \mid \theta_1, ..., \theta_i \in (\tau_1 \to \tau_2)^{atom}\}$.

We will usually use the letter θ to represents atomic mappings, and the letter σ to represent conjunctive mappings. Given a conjunctive mapping σ (resp. an atomic mapping θ) and a type τ, we write $\sigma :: \tau$ (resp. $\theta ::_a \tau$) the relation $\sigma \in (\tau)^\wedge$ (resp. $\theta \in (\tau)^{atom}$). For the sake of simplicity, we identify the atomic mapping θ with the conjunctive mapping $\bigwedge\{\theta\}$.

Given a term t and a conjunctive mapping σ, we define a judgment as a tuple $\Theta \vdash t \rhd \sigma$, pronounced "from the environment Θ, one can prove that t matches the conjunctive mapping σ", where the environment Θ is a partial mapping from $\mathcal{V} \uplus \mathcal{N}$ to conjunctive mapping. Given an environment Θ, $\alpha \in \mathcal{V} \uplus \mathcal{N}$ and a conjunctive mapping σ, we define the environment $\Theta' = \Theta, \alpha \rhd \sigma$ as $Dom(\Theta') = Dom(\Theta) \cup \{\alpha\}$ and $\Theta'(\alpha) = \sigma$ if $\alpha \notin Dom(\Theta)$, $\Theta'(\alpha) = \sigma \wedge \Theta(\alpha)$ otherwise, and $\Theta'(\beta) = \Theta(\beta)$ if $\beta \neq \alpha$.

We define the following judgement rules:

$$\frac{\Theta \vdash t \triangleright \theta_1 \quad ... \quad \Theta \vdash t \triangleright \theta_n}{\Theta \vdash t \triangleright \bigwedge\{\theta_1, ..., \theta_n\}}(Set) \qquad \frac{}{\Theta, \alpha \triangleright \bigwedge\{\theta_1, ..., \theta_n\} \vdash \alpha \triangleright \theta_i}(At) \ \textit{(for all i)}$$

$$\frac{}{\Theta \vdash a \triangleright \sigma_1 \to ... \to \sigma_{i \leq arity(a)} \to q_\infty}(\Sigma) \ \textit{(for } a \in \Sigma \textit{ and } \exists j \ \sigma_j = q_\infty)$$

$$\frac{\Theta \vdash t_1 \triangleright \sigma \to \theta \quad \Theta \vdash t_2 \triangleright \sigma}{\Theta \vdash t_1 \ t_2 \triangleright \theta}(App)$$

$$\frac{}{\Theta \vdash t \triangleright q_\infty \to q_\infty}(q_\infty \to q_\infty) \ \textit{(if } t : \tau_1 \to \tau_2) \qquad \frac{\Theta \vdash t_1 \triangleright q_\infty}{\Theta \vdash t_1 \ t_2 \triangleright q_\infty}(q_\infty)$$

Remark that there is no rules that directly involves $q_\perp$, but it does not mean that no term matches $q_\perp$, since it can appear in Θ. Rules like (At) or (App) may be used to state that a term matches $q_\perp$.

The main property of IO derivation is captured by the rules (q_∞) and $(q_\infty \to q_\infty)$, which describes the fact that if a term t contains a subterm whose maximal IO derivations are infinite, then maximal IO derivations of t are also infinite. This is not the case for OI derivations.

We say that $(\mathcal{G}, t)$ matches the conjunctive mapping σ written $\vdash (\mathcal{G}, t) \triangleright \sigma$ if there exists an environment Θ, called a witness environment of $\vdash (\mathcal{G}, t) \triangleright \sigma$, such that (1) $Dom(\Theta) = \mathcal{N}$, (2) $\forall F : \tau \in \mathcal{N},\ \Theta(F) :: \tau$, (3) if $F\ x_1...x_k \to e\ \in \mathcal{R}$ and $\Theta \vdash F \triangleright \sigma_1 \to ... \to \sigma_{i\leq k} \to q$ then either there exists j such that $q_\infty \in \sigma_j$, or $i = k$ and $\Theta, x_1 \triangleright \sigma_1, ..., x_k \triangleright \sigma_k \vdash e \triangleright q$, (4) $\Theta \vdash t \triangleright \sigma$.

The following two results state that this type system matches the properties $\mathcal{P}_\perp$ and $\mathcal{P}_\infty$ and furthermore we can construct an universal environment, $\Theta^\star$, that can correctly judge any term.

Theorem 3 (Soundness and Completeness). *Let $\mathcal{G}$ be an HORS, and t be term (of any type), $\vdash (\mathcal{G}, t) \triangleright q_\infty$ (resp. $\vdash (\mathcal{G}, t) \triangleright q_\perp$) if and only if $\mathcal{P}_\infty(t)$ (resp. $\mathcal{P}_\perp(t)$) holds.*

Proposition 1 (Universal Witness). *There exists an environment $\Theta^\star$ such that for all term t, the judgment $\vdash (\mathcal{G}, t) \triangleright \sigma$ holds if and only if $\Theta^\star \vdash t \triangleright \sigma$.*

Proof (Sketch). To compute $\Theta^\star$, we start with an environment Θ_0 satisfying Properties (1) and (2) ($Dom(\Theta_0) = \mathcal{N}$ and $\forall F : \tau \in \mathcal{N},\ \Theta_0(F) :: \tau$) that is able to judge any term $t : \tau$ with any conjunctive mapping $\sigma :: \tau$.

Then let $\mathcal{F}$ be the mapping from the set of environments to itself, such that for all $F : \tau_1 \to ... \to \tau_k \to o \in \mathcal{N}$, if $F\ x_1...x_k \to e \in \mathcal{R}$ then,

$$\begin{aligned}\mathcal{F}(\Theta)(F) = \{\sigma_1 \to ... \to \sigma_k \to q \mid\ & q \in Q \wedge \forall i\ \sigma_i :: \tau_i \\ & \wedge \Theta, x_1 \triangleright \sigma_1, ..., x_k \triangleright \sigma_k \vdash e : q\} \\ \cup \{\sigma_1 \to ... \to \sigma_{i\leq k} \to q_\infty \mid\ & \wedge \forall i\ \sigma_i :: \tau_i \wedge \exists j\ q_\infty \in \sigma_j\} \\ \cup \{\sigma_1 \to ... \to \sigma_k \to q_\perp \mid\ & \forall i\ \sigma_i :: \tau_i \wedge \exists j\ q_\infty \in \sigma_j\}\ .\end{aligned}$$

We iterate $\mathcal{F}$ until we reach a fixpoint. The environment we get is $\Theta^\star$, it verifies properties (1) (2) and (3). Furthermore we can show that this is the maximum of all environments satisfying these properties, i.e. if $\vdash (\mathcal{G}, t) \triangleright \sigma$ then $\Theta^\star \vdash t \triangleright \sigma$. $\square$

4.2 Self-Correcting Scheme

For all term $t : \tau \in \mathcal{T}(\Sigma \uplus \mathcal{N})$, we define $[\![t]\!] \in (\tau)^\wedge$, called the witnessed value of t, as the conjunction of all atomic mappings θ such that $\Theta^\star \vdash t \triangleright \theta$ (recall that $\Theta^\star$ is the environment of Proposition 1). In particular $\mathcal{P}_\perp(t)$ (resp. $\mathcal{P}_\infty(t)$) holds if and only if $q_\perp \in [\![t]\!]$ (resp. $q_\infty \in [\![t]\!]$).

Given two terms $t_1 : \tau' \to \tau$ and $t_2 : \tau'$ the only rules we can apply to judge $\Theta^\star \vdash t_1\ t_2 \triangleright \theta$ are (App), $(q_\infty \to q_\infty)$ and (q_∞) (recall that θ is atomic). Then, we see that θ only depends on which atomic mappings are matched by t_1 and t_2. In other words $[\![t_1\ t_2]\!]$ only depends on $[\![t_1]\!]$ and $[\![t_2]\!]$. Hence we can associate to $[\![t_1]\!]$ a function $f_{[\![t_1]\!]}$ defined on $\{\sigma \mid \exists t'_2\ [\![t'_2]\!] = \sigma\}$ such that $f_{[\![t_1]\!]}(\sigma) = [\![t_1\ t'_2]\!]$ with $[\![t'_2]\!] = \sigma$. We use the notation $[\![t_1]\!]\ [\![t_2]\!]$ as a shortcut for $f_{[\![t_1]\!]}([\![t'_2]\!])$.

In this section, given a scheme $\mathcal{G} = \langle \mathcal{V}, \Sigma, \mathcal{N}, \mathcal{R}, S\rangle$, we transform it into $\mathcal{G}' = \langle \mathcal{V}', \Sigma, \mathcal{N}', \mathcal{R}', S\rangle$ which is basically the same scheme except that while it is producing an IO derivation, it evaluates $[\![t']\!]$ for any subterm t' of the current term and label t' with $[\![t']\!]$. Note that if $t \to_{\text{IO}} t'$, then $[\![t]\!] = [\![t']\!]$. Since we cannot syntactically label terms, we will label all symbols by the witnessed value of their arguments, for example if the term $F\ t_1 ... t_k$, we will label F with the k-tuple $([\![t_1]\!], ..., [\![t_k]\!])$.

A problem may appear if some of the arguments are not fully applied, for example imagine we want to label $F\ H$ with $H : o \to o$. We will label F with $[\![H]\!]$, but since H has no argument we do not know how to label it. The problem is that we cannot wait to label it because once a non-terminal is created, the derivation does not deal explicitly with it. The solution is to create one copy of H per possible witnessed value for its argument (here there are four of them: $\bigwedge\{\}, \bigwedge\{q_\perp\}, \bigwedge\{q_\infty\}, \bigwedge\{q_\perp, q_\infty\}$). This means that $F^{[\![H]\!]}$ would not have the same type as F: F has type $(o \to o) \to o$, but $F^{[\![H]\!]}$ will have type $(o \to o)^4 \to o$. Hence, $F\ H$ will be labelled the following way: $F^{[\![H]\!]}\ H^{\bigwedge\{\}}\ H^{\bigwedge\{q_\perp\}}\ H^{\bigwedge\{q_\infty\}}\ H^{\bigwedge\{q_\perp, q_\infty\}}$. Note that even if F has 4 arguments, it only has to be labelled with one witnessed value since all four arguments represent different labelling of the same term. We now formalize these notions.

Let us generalize the notion of witnessed value to deal with terms containing some variables. Given an environment on the variables $\Theta^\mathcal{V}$ such that $Dom(\Theta^\mathcal{V}) \subseteq \mathcal{V}$ and if $x : \tau$ then $\Theta^\mathcal{V}(x) :: \tau$, and given a term $t : \tau \in \mathcal{T}(\Sigma \uplus \mathcal{N} \uplus Dom(\Theta^\mathcal{V}))$, we define $[\![t]\!]_{\Theta^\mathcal{V}} \in (\tau)^\wedge$, as the conjunction of all atomic mappings θ such that $\Theta^\star, \Theta^\mathcal{V} \vdash t \triangleright \theta$. Given two terms $t_1 : \tau' \to \tau$ and $t_2 : \tau'$ we still have that $[\![t_1\ t_2]\!]_{\Theta^\mathcal{V}}$ only depends on $[\![t_1]\!]_{\Theta^\mathcal{V}}$ and $[\![t_2]\!]_{\Theta^\mathcal{V}}$.

To a type $\tau = \tau_1 \to ... \to \tau_k \to o$ we associate the integer

$$n_\tau = Card\big(\{(\sigma_1, ..., \sigma_k) \mid \forall i\ \sigma_i \in (\tau_i)^\wedge\}\big)$$

and a complete ordering of $\{(\sigma_1, ..., \sigma_k) \mid \forall i\ \sigma_i \in (\tau_i)^\wedge\}$ denoted vec^τ_1, vec^τ_2, ... , $vec^\tau_{n_\tau}$. We define inductively the type $\tau^+ = (\tau_1^+)^{n_{\tau_1}} \to ... \to (\tau_k^+)^{n_{\tau_k}} \to o$.

To a non terminal $F : \tau_1 \to ... \to \tau_k \to o$ (resp. a variable $x : \tau_1 \to ... \to \tau_k \to o$) and a tuple $\sigma_1 :: \tau_1, ..., \sigma_k :: \tau_k$, we associate the non-terminal $F^{\sigma_1,...,\sigma_k} : \tau_1^{n_{\tau_1}} \to ... \to \tau_k^{n_{\tau_k}} \to o \in \mathcal{N}'$ (resp. a variable $x^{\sigma_1,...,\sigma_k} : \tau_1^{n_{\tau_1}} \to ... \to \tau_k^{n_{\tau_k}} \to o \in \mathcal{V}'$).

Given a term $t : \tau = \tau_1 \to ... \to \tau_k \to o \in \mathcal{T}(\mathcal{V} \uplus \Sigma \uplus \mathcal{N})$ and an environment on the variables $\Theta^\mathcal{V}$ such that $Dom(\Theta^\mathcal{V}) \subseteq \mathcal{V}$ contains all the variables appearing in t, we define inductively the term $t_{\Theta^\mathcal{V}}^{+\sigma_1,...,\sigma_k} : \tau^+ \in \mathcal{T}(\mathcal{V}' \uplus \Sigma' \uplus \mathcal{N}')$ for all $\sigma_1 :: \tau_1, ..., \sigma_k :: \tau_k$. If $t = F \in \mathcal{N}$ (resp. $t = x \in \mathcal{V}$), $t_{\Theta^\mathcal{V}}^{+\sigma_1,...,\sigma_k} = F^{\sigma_1,...,\sigma_k}$ (resp. $t_{\Theta^\mathcal{V}}^{+\sigma_1,...,\sigma_k} = x^{\sigma_1,...,\sigma_k}$), if $t = a \in \Sigma$, $t_{\Theta^\mathcal{V}}^{+\sigma_1,...,\sigma_k} = a$. Finally consider the case where $t = t_1\ t_2$ with $t_1 : \tau' \to \tau$ and $t_2 : \tau'$. Let $\sigma = [\![t_2]\!]_{\Theta^\mathcal{V}}$. Remark that $t_1{}_{\Theta^\mathcal{V}}^{+\sigma,\sigma_1,...,\sigma_k} : (\tau'^+)^{n_{\tau'}} \to \tau^+$. We define $(t_1\ t_2)_{\Theta^\mathcal{V}}^{+\sigma_1,...,\sigma_k} = t_1{}_{\Theta^\mathcal{V}}^{+\sigma,\sigma_1,...,\sigma_k}\ t_2{}_{\Theta^\mathcal{V}}^{+vec_1^{\tau'}} ...\ t_2{}_{\Theta^\mathcal{V}}^{+vec_{n_{\tau'}}^{\tau'}}$. Note that since this transformation is only duplicating and anotating, given a term $t^{+\sigma_1,...,\sigma_k}$ we can uniquely find the unique term t associated to it.

Let $F : \tau_1 \to ... \to \tau_k \to o \in \mathcal{N}$, $\sigma_1 :: \tau_1, ..., \sigma_k :: \tau_k$, and $\Theta^\mathcal{V} = x_1 \triangleright \sigma_1, ..., x_k \triangleright \sigma_k$. If $F\ x_1 ... x_k\ \to\ e \in \mathcal{R}$, we define in $\mathcal{R}'$ the rule

$$F^{\sigma_1,...,\sigma_k}\ x_1^{+vec_1^{\tau_1}} ...\ x_1^{+vec_{n_{\tau_1}}^{\tau_1}}\ ...\ x_k^{+vec_1^{\tau_k}} ...\ x_k^{+vec_{n_{\tau_k}}^{\tau_k}}\ \to\ e_{\Theta^\mathcal{V}}^+ .$$

Finally, recall that $\mathcal{G}' = \langle \mathcal{V}', \Sigma, \mathcal{N}', \mathcal{R}', S \rangle$.

The following theorem states that $\mathcal{G}'$ is just a labeling version of $\mathcal{G}$ and that it acts the same.

Theorem 4 (Equivalence between $\mathcal{G}$ and $\mathcal{G}'$). *Given a term $t : o$, $\|\mathcal{G}'_{t^+}\|_{\text{IO}} = \|\mathcal{G}_t\|_{\text{IO}}$.*

We transform $\mathcal{G}'$ into the scheme $\mathcal{G}''$ that will directly turn into $\perp$ a redex t such that $q_\perp \in [\![t]\!]$. For technical reason, instead of adding $\perp$ we add a non terminal $Void : o$ and a rule $Void \to Void$. $\mathcal{G}' = \langle \mathcal{V}', \Sigma, \mathcal{N}' \uplus \{Void : o\}, \mathcal{R}'', S \rangle$ such that $\mathcal{R}''$ contains the rule $Void \to Void$ and for all $F \in \mathcal{N}$, if $q_\perp \in [\![F]\!]\ \sigma_1\ ...\ \sigma_k$ then

$$F^{\sigma_1,...,\sigma_k}\ x_1^{+vec_1^{\tau_1}} ...\ x_1^{+vec_{n_{\tau_1}}^{\tau_1}} ...\ x_k^{+vec_1^{\tau_k}} ...\ x_k^{+vec_{n_{\tau_k}}^{\tau_k}} \to Void$$

otherwise we keep the rule of $\mathcal{R}'$.

The next theorem concludes Section 4.

Theorem 5 (IO vs OI). *Let $\mathcal{G}$ be a higher-order recursion scheme. Then one can construct a scheme G'' having the same order of $\mathcal{G}$ such that $\|\mathcal{G}''\| = \|\mathcal{G}\|_{\text{IO}}$.*

Proof (Sketch). First, given a term $t : o$, one can prove that $\|\mathcal{G}''_{t^+}\|_{\text{IO}} = \|\mathcal{G}'_{t^+}\|_{\text{IO}}$.

Then take a redex t such that $\|\mathcal{G}''_t\|_{\text{IO}} = \perp$, i.e. $q_\perp \in [\![\mathcal{G}_t]\!]$. There is only one OI derivation from t: $t \to Void \to Void \to ...$, then $\|\mathcal{G}''_t\| = \perp$. We can extend this result saying that if there is the symbol $\perp$ at node u in $\|\mathcal{G}''_t\|_{\text{IO}}$, then there is $\perp$ at node u in $\|\mathcal{G}''_t\|$. Hence, since $\|\mathcal{G}''_t\|_{\text{IO}} \sqsubseteq \|\mathcal{G}''_t\|$, we have $\|\mathcal{G}''\| = \|\mathcal{G}''\|_{\text{IO}}$. Then $\|\mathcal{G}''\| = \|\mathcal{G}''\|_{\text{IO}} = \|\mathcal{G}'\|_{\text{IO}} = \|\mathcal{G}\|_{\text{IO}}$. $\square$

5 Avoiding Non Productive OI Derivations

The technics shown in Section 4 may seem quite heavy compared to the lazying transformation in Section 3. They are actually far more general. In this section, we show how they can be used to enhance OI derivations of a scheme.

Given an infinite OI derivation $t_1 \rightarrow_{\text{OI}} t_2 \rightarrow_{\text{OI}} ...$, we say that it is ultimately non productive if at some point the tree associated to it, does not increase, i.e. if there exists i such that for all $j \geq i$, $(t_j)^{\perp} = (t_i)^{\perp}$.

Example 6. Take the scheme $\langle\{\mathtt{a} : o\}, \{S : o, H : o \rightarrow o\}, \{x : o\}, S, \{S \rightarrow H\ \mathtt{a}\ ,\ H\ x \rightarrow H\ (H\ x)\}\rangle$. The OI derivation $S \rightarrow H\ \mathtt{a} \rightarrow_{\text{OI}} H\ (H\ \mathtt{a}) \rightarrow_{\text{OI}} H\ (H\ (H\ \mathtt{a})) \rightarrow_{\text{OI}} ...$ is ultimately non productive, because its associated tree sequence is $\perp, \perp, \perp, ...$. In this example, this is the only OI derivation from the term $H\ \mathtt{a}$. Now add to the scheme a terminal symbol $\mathtt{f} : o^2 \rightarrow o$, a non terminal symbol $F : o \rightarrow$ and the rule $F\ x \rightarrow x\ (F\ x)$. the OI derivation $\mathtt{f}\ (F\ \mathtt{a})\ (H\ \mathtt{a}) \rightarrow_{\text{OI}} \mathtt{f}\ (F\ \mathtt{a})\ (H\ (H\ \mathtt{a})) \rightarrow_{\text{OI}} \mathtt{f}\ (F\ \mathtt{a})\ (H\ (H\ (H\ \mathtt{a}))) \rightarrow_{\text{OI}} ...$ is ultimately non productive, however there exists a productive derivation from $\mathtt{f}\ (F\ \mathtt{a})\ (H\ \mathtt{a})$: $\mathtt{f}\ (F\ \mathtt{a})\ (H\ \mathtt{a}) \rightarrow_{\text{OI}} \mathtt{f}\ (\mathtt{a}\ (F\ \mathtt{a}))\ (H\ \mathtt{a}) \rightarrow_{\text{OI}} \mathtt{f}\ (\mathtt{a}\ (\mathtt{a}\ (F\ \mathtt{a})))\ (H\ \mathtt{a}) \rightarrow_{\text{OI}} ...$.

Using the same tools as those in Section 4, it is possible to transform a scheme $\mathcal{G}$ into a scheme $\mathcal{G}''$ that produce the same valuation tree, in which no ultimately non productive OI derivation is possible. The construction proceeds the same way as in section 4, first we create a type system to decide wether a term would lead to $\perp$ with OI derivations, then we embed this type system into the scheme, finally we make the scheme turn to $\perp$ all the redexes that would lead to $\perp$ in the valuation tree.

6 Conclusion

We have shown that value trees obtained from schemes using innermost-outermost derivations (IO) are the same as those obtained using unrestricted derivations. More precisely given an order-n scheme G we create an order-$(n+1)$ scheme $\overline{\mathcal{G}}$ such that $\|\overline{\mathcal{G}}\|_{IO} = \|\mathcal{G}\|$. However, the increase of the order seems unavoidable. We also create an order-n scheme $\mathcal{G}''$ such that $\|\mathcal{G}''\| = \|\mathcal{G}\|_{IO}$. In this case the order does not increase, however the size of the scheme blows up while it remains almost the same in $\overline{\mathcal{G}}$. In the last section, we have given a glimpse on how the technics shown in Section 4 can be used to treat other problems.

References

1. K. Aehlig. A finite semantics of simply-typed lambda terms for infinite runs of automata. In *"Proc. of Computer Science Logic, 20th Annual Conference of the EACSL"*, volume 4207 of *Lecture Notes in Comput. Sci.*, pages 104–118. Springer-Verlag, 2006.
2. B. Courcelle. A representation of trees by languages I. *Theoret. Comput. Sci.*, 6:255–279, 1978.

3. B. Courcelle. A representation of trees by languages II. *Theoret. Comput. Sci.*, 7:25–55, 1978.
4. B. Courcelle and M. Nivat. The algebraic semantics of recursive program schemes. In *Proc. 7th Symposium, Mathematical Foundations of Computer Science 1978*, volume 64 of *Lecture Notes in Comput. Sci.*, pages 16–30. Springer-Verlag, 1978.
5. W. Damm. Higher type program schemes and their tree languages. In *Theoretical Computer Science, 3rd GI-Conference*, volume 48 of *Lecture Notes in Comput. Sci.*, pages 51–72. Springer-Verlag, 1977.
6. W. Damm. Languages defined by higher type program schemes. In *Proc. 4th Colloq. on Automata, Languages, and Programming (ICALP)*, volume 52 of *Lecture Notes in Comput. Sci.*, pages 164–179. Springer-Verlag, 1977.
7. W. Damm. The IO- and OI-hierarchies. *Theoret. Comput. Sci.*, 20:95–207, 1982.
8. J. Engelfriet and E. M. Schmidt. IO and OI. I. *J. Comput. System Sci.*, 15(3):328–353, 1977.
9. J. Engelfriet and E. M. Schmidt. IO and OI. II. *J. Comput. System Sci.*, 16(1):67–99, 1978.
10. M. Hague, A. S. Murawski, C.-H. L. Ong, and O. Serre. Collapsible pushdown automata and recursion schemes. In *Proceedings of the 23rd Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 452–461. IEEE Computer Society, 2008.
11. K. Indermark. Schemes with recursion on higher types. In *Proc. 5th Symposium, Mathematical Foundations of Computer Science 1976*, volume 45 of *Lecture Notes in Comput. Sci.*, pages 352–358. Springer-Verlag, 1976.
12. T. Knapik, D. Niwiński, and P. Urzyczyn. Higher-order pushdown trees are easy. In *Proceedings of the 5th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS)*, volume 2303 of *Lecture Notes in Comput. Sci.*, pages 205–222. Springer-Verlag, 2002.
13. N. Kobayashi. Types and higher-order recursion schemes for verification of higher-order programs. In *Proc. 36th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*, pages 416–428. ACM, 2009.
14. N. Kobayashi and C.-H. L. Ong. A type system equivalent to the modal mu-calculus model checking of higher-order recursion schemes. In *Proceedings of the 24th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 179–188. IEEE Computer Society, 2009.
15. M. Nivat. On the interpretation of recursive program schemes. In *Symposia Matematica*, 1972.

A From OI to IO

Complement of Definitions

A n holes context is a term $C[\bullet_1^{\tau_1}, ..., \bullet_n^{\tau_n}] \in \mathcal{T}(\Gamma \uplus \{\bullet_i^{\tau_i} : \tau_i \mid 1 \leq i \leq n\})$ containing exactly one occurrence of $\bullet_i$ for all i. (We will generally omit to write the type τ_i in the notation $\bullet_i^{\tau_i}$).

For all $i_1, ..., i_k \leq n$ we are interested in the application

$$t_1, ..., t_k \mapsto \big(C[\bullet_1, ..., \bullet_n]\big)_{[\forall j \leq k \ \bullet_{i_j} \mapsto t_j]}$$

with $t_j \in \mathcal{T}_{\tau_{i_j}}(\Gamma)$ for all j. (notice that the order of the substitution is not important). One can consider $\big(C[\bullet_1, ..., \bullet_n]\big)_{[\forall j \leq k \ \bullet_{i_j} \mapsto t_j]}$ as a $n-k$ holes context. We may write $C[\bullet_1]...[\bullet_n]$ to denote the context $C[\bullet_1, ..., \bullet_n]$ and extend this notation to $\big(C[\bullet_1, ..., \bullet_n]\big)_{[\forall j \leq k \ \bullet_{i_j} \mapsto t_j]}$, for example, given the context $C[\bullet_1][\bullet_2][\bullet_3]$, we define $C[t_1][\bullet_2][t_3] = \big(C[\bullet_1][\bullet_2][\bullet_3]\big)_{[\bullet_1 \mapsto t_1, \bullet_3 \mapsto t_3]}$

Given a one hole context $C[\bullet]$, we define inductively the ***head symbol sequence*** $\mathrm{hss}(C)$ which is a finite sequence of symbols of Γ: if $C[\bullet] = \bullet$, then $\mathrm{hss}(C)$ is the empty sequence, if $C[\bullet] = \alpha \ t_1, ..., t_{i-1} C'[\bullet] t_{i+1} ... t_k$, then $\mathrm{hss}(C) = \alpha, \mathrm{hss}(C')$.

Proposition 2. *Given a n holes context $C[\bullet_1]...[\bullet_n]$, for all i, for all $t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n$ and $s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n$:*
$\mathrm{hss}(C[t_1]...[t_{i-1}][\bullet][t_{i+1}]...[t_n]) = \mathrm{hss}(C[s_1]...[s_{i-1}][\bullet][s_{i+1}]...[s_n])$

Proof (Proposition 2). We prove this proposition by induction on the size of the context, for all n . If $C[\bullet_i] = \bullet_i$ is a 1 hole context, then the result is prooven. If $C[\bullet_1]...[\bullet_n] = a \ t_1 ... t_n$ there exists exactly one k such that t_k contains one ocurrence of $\bullet_i$, if we look all the occurrences of some $\bullet_j$ in t_k, we can state that t_k is a l holes context $C'[\bullet_{j_1}]...[\bullet_{j_l}]$for some l. Moreover

$$\mathrm{hss}(C[t_1]...[t_{i-1}][\bullet][t_{i+1}]...[t_n]) = a, \mathrm{hss}(C'[t_{j_1}]...[\bullet_i][t_{j_l}]) \text{ and}$$

$$\mathrm{hss}(C[s_1]...[s_{i-1}][\bullet][s_{i+1}]...[s_n]) = a, \mathrm{hss}(C'[s_{j_1}]...[\bullet_i][s_{j_l}])$$

Since $\mathrm{hss}(C'[t_{j_1}]...[\bullet_i][t_{j_l}]) = \mathrm{hss}(C'[s_{j_1}]...[\bullet_i][s_{j_l}])$ by hypothesis of induction, we have

$$\mathrm{hss}(C[t_1]...[t_{i-1}][\bullet][t_{i+1}]...[t_n]) = \mathrm{hss}(C[s_1]...[s_{i-1}][\bullet][s_{i+1}]...[s_n])$$

Proposition 3. *Let t be a term, and let $C_1[\bullet]$ and $C_2[\bullet]$ be two contexts such that $t = C_1[t_1]$ and $t = C_2[t_2]$, then: either there exists a context C_1' such that $C_2[\bullet] = C_1[C_1'[\bullet]]$ (1), Or there exists a context C_2' such that $C_1[\bullet] = C_2[C_2'[\bullet]]$ (2), or there exists a two hole context $C[\bullet_1][\bullet_2]$ such that $C[t_1][\bullet] = C_2[\bullet]$ and $C[\bullet][t_2] = C_1[\bullet]$ (3).*

Proof (Proposition 3). We proceed by induction. If $C_1[\bullet] = \bullet$, then $C_2[\bullet] = C_1[C_2[\bullet]]$, in the same way, if $C_2[\bullet] = \bullet$, $C_1[\bullet] = C_2[C_1[\bullet]]$. Else $C_1[\bullet] = a\ s_1...s_{i-1}C'_1[\bullet]s_{i+1}...s_k$ and $C_2[\bullet] = a\ s_1...s_{j-1}C'_2[\bullet]s_{j+1}...s_k$. If $i \neq j$ (we assume w.l.o.g. that $i < j$) we set $C[\bullet_1][\bullet_2] = a\ s_1...s_{i-1}C'_1[\bullet_1]s_{i+1}...s_{j-1}C'_2[\bullet_2]s_{j+1}...s_k$, we have that $C[t_1][\bullet] = C_2$ and $C[\bullet][t_2] = C_1[\bullet]$. If $i = j$, then $C'_1[t_1] = C'_2[t_2]$, by induction:

- Either there exists a two hole context $C'[\bullet_1][\bullet_2]$ such that $C'[t_1][\bullet] = C'_2[\bullet]$ and $C'[\bullet][t_2] = C'_1[\bullet]$, in that case we set $C[\bullet_1][\bullet_2] = a\ s_1...s_{i-1}C'[\bullet_1][\bullet_2]s_{i+1}...s_k$, and we have $C[t_1][\bullet] = C_2[\bullet]$ and $C[\bullet][t_2] = C_1[\bullet]$,
- Or there exists a context C''_1 such that $C'_2[\bullet] = C'_1[C''_1[\bullet]]$, in that case $C_2[\bullet] = C_1[C''_1[\bullet]]$.
- Or there exists a context C''_2 such that $C'_1[\bullet] = C'_2[C''_2[\bullet]]$, in that case $C_1[\bullet] = C_2[C''_2[\bullet]]$.

Correctness of the Transformation

We remark that $\overline{t}$ does not contain any occurrence of Δ, I or δ. We also remark that any subterms of $\overline{t}$ has type $\overline{\tau}$ for some type τ, hence it can not have ground type and in particular, it is not a redex. Moreover, given two terms t_1, t_2, if the term $\overline{t_1}\ \overline{t_2}$ is valid, then it is equal to $\overline{t_1\ t_2}$, in particular, it is an "overlined" term. It follows by induction, that given three terms t, t_1, t_2, if the term $\overline{t}_{[\overline{t}_1 \mapsto \overline{t}_2]}$ is well defined, then it is equal to $\overline{t_{[t_1 \mapsto t_2]}}$ wich is also well defined.

Proof (Lemma 1). First We need the following claim.

Claim ($\star$). For all accessible term t (with unrestricted derivations), for all context $C[\bullet]$ such that $t = C[red]$ with red being a redex , $hss(C)$ only contains terminals symbols. Furthermore, the term red doesn't contain occurence of any terminal symbol.

Proof (Claim ($\star$)). We prove it by induction. I satisfies Claim A, $\overline{S}\ \Delta$ too. Assume that $t = C[\overline{F}\ t_1...t_k]$ satisfies Claim A, with $k = \text{arity}(\overline{F})$ and $F \in \mathcal{N}$. If $\overline{F}\ x_1...x_k \to_{\overline{G}} \overline{e}\Delta \in \overline{\mathcal{R}}$, let $t' = C[\overline{e}_{[\forall i\ x_i \mapsto t_i]}\Delta]$. Let $C'[\bullet]$ a context and $red = \gamma\ s_1...s_{\text{arity}(\gamma)}$ with $\gamma \in \overline{\mathcal{N}}$ a redex such that $t' = C'[red]$.
First, we notice that since $\overline{e}_{[\forall i\ x_i \mapsto t_i]}\ \Delta$ is a ground type term only containing non-terminal symbols, it is a redex , let $\rho\ r_1...r_{\text{arity}(\rho)-1}\ \Delta = \overline{e}_{[\forall i\ x_i \mapsto t_i]}\ \Delta$.

Using Proposition 3 we now that there are four options:

1. either $C[\bullet] = C'[\gamma\ s_1...s_{i-1}C''[\bullet]s_{i+1}...s_{\text{arity}(\gamma)}]$ with C'' a context,
2. or $C'[\bullet] = C[\rho\ r_1...r_{i-1}C''[\bullet]r_{i+1}...r_{\text{arity}(\rho)-1}\Delta]$ with C'' a context,
3. or $C'[\bullet] = C[\bullet]$,
4. or there exists a two holes context $\mathbb{C}[\bullet_1][\bullet_2]$ such that $C[\bullet] = \mathbb{C}[\bullet][red]$ and $C'[\bullet] = \mathbb{C}[\overline{e}_{[\forall i\ x_i \mapsto t_i]}\Delta][\bullet]$.

Option 1 is impossible, otherwise γ would be an element of $hss(C)$.
Option 2 would imply that $\overline{e}_{[\forall i\ x_i \mapsto t_i]} = \rho\ r_1...r_{i-1}C''[red]r_{i+1}...r_{\text{arity}(\rho)-1}$ then $\overline{e}_{[\forall i\ x_i \mapsto t_i]}$ contains a ground typed term, which can't be true, see Remark A.

If **Option 3** is true. Then $\mathrm{hss}(C') = \mathrm{hss}(C)$ which by induction only contains terminal symbols. Since there is no terminal symbols in $\overline{e}$ and in t_i for all i, there is no terminal in $\overline{e}\Delta = red$. Hence t' satisfies Claim A.
If **Option 4** is true, then $t = \mathbb{C}[\overline{F}\, t_1...t_n][red]$. Then by induction, $\mathrm{hss}(\mathbb{C}[\overline{F}\, t_1...t_n][\bullet])$ only contains terminal symbols. But, using Proposition 2, we know that

$$\mathrm{hss}(C'[\bullet]) = \mathrm{hss}(\mathbb{C}[\overline{e}_{[\forall i\ x_i \mapsto t_i]}\Delta][\bullet]) = \mathrm{hss}(\mathbb{C}[\overline{F}\ t_1...t_n][\bullet]).$$

Then $\mathrm{hss}(C'[\bullet])$ only contains terminal symbols. Furthermore, since red is a subterm of t, by induction it only contains non-terminal symbols, which proves that t' satisfies Claim $(\star)$.

Assume that $t = C[\overline{a}\ t_1...t_k]$ satisfies Claim A with $a \in \Sigma$ and $k = \mathrm{arity}(a)$, and let $t' = C[a\ (t_1\Delta)...(t_k\Delta)]$. We can prove that t' satisfies Claim A in a similar way).

Let $t = C[red]$ an accessible term with red a redex , let exp be the rewrite expression of red, and let's look at the derivation $t = C[red] \rightarrow_{\overline{G}} t' = C[exp]$.

Claim A tells us that $\mathrm{hss}(C)$ only contains terminals, hence there is no redex containing an occurence of $\bullet$ in C, hence the derivation is OI.Assume that $red = \gamma\ t_1...t_{i-1}C'(t)t_{i+1}...t_{\mathrm{arity}(\gamma)}$ with C' a context and t a term.

Then $t = C[\gamma\ t_1...t_{i-1}C'(t)t_{i+1}...t_{\mathrm{arity}(\gamma)}]$ then $\mathrm{hss}(C[\gamma\ t_1...t_{i-1}C'(\bullet)t_{i+1}...t_{\mathrm{arity}(\gamma)}])$ contains a non terminal symbol γ, hence Claim A tells that t is not a redex , so no non-trivial subterm of red is a redex , so the derivation is IO.

Proof (Theorem 1). Lemma 1 shows that we only have to prove that $\|G\|_{OI} = \|\overline{G}\|$. Concretely we will show that:

$$\forall t \in \mathrm{Acc}_G,\ \exists t' \in \mathrm{Acc}_{\overline{G}} :\ t^{\perp} \sqsubseteq (t')^{\perp} \qquad (1)$$

$$\forall t' \in \mathrm{Acc}_{\overline{G}},\ \exists t \in \mathrm{Acc}_G :\ (t')^{\perp} \sqsubseteq t^{\perp} \qquad (2)$$

Definition 1 **($\|\cdot\|$-transformation).** *We define inductively the transformation* $\|\cdot\| : \mathcal{T}^o(\Sigma \uplus \mathcal{N}) \rightarrow \mathcal{T}^o(\Sigma \uplus \overline{\mathcal{N}})$ *:*

- $\|a\ t_1...t_{arity(a)}\| = a\ \|t_1\|...\|t_{arity(a)}\|$ *for all* $a \in \Sigma$*,*
- $\|red\| = \overline{red}\ \Delta$ *for red a redex .*

Remark 1. Notice that $t^{\perp} = (\|t\|)^{\perp}$.

Claim (♣). If $t \in \mathcal{T}^o(\Sigma \uplus \mathcal{N})$ then $\overline{t}\ \Delta\ \rightarrow_{\overline{G}}\ \|t\|$.

Proof (Claim (♣)). The proof is done by induction. If t is a redex then $\|t\| = \overline{t}\ \Delta$. If $t = a\ t_1...t_k$ with $a \in \Sigma$ and $k = \mathrm{arity}(a)$, assume that for all i, $\overline{t_i}\Delta \rightarrow \|t_i\|$. We have $\overline{t}\ \Delta = \overline{a}\ \overline{t_1}...\overline{t_k}$ so $\overline{t} \rightarrow_{\overline{G}} a\ (\overline{t_1}\Delta)...(\overline{t_k}\Delta)$. Hence $\overline{t}\Delta \rightarrow^*_{\overline{G}} a\ \|t_1\|...\|t_k\| = \|t\|$.

Claim (♡). For all t, if $t \in Acc_G$, then $\|t\| \in Acc_{\overline{G}}$. **This claim implies property** (1).

Proof (Proof of Claim (♡)). We prove this by induction. If $t = S$, $\|t\| = S\ \Delta$, and $I \to S\ \Delta$, so $\|t\| \in Acc_{\overline{G}}$.

Let $t = C[F\ t_1...t_k] \in \mathrm{Acc}_G$ with $k = \mathrm{arity}(F)$ and $\mathrm{hss}(C)$ contains ony terminal symbols. Assume that $\|t\| \in \mathrm{Acc}_G$. Given that $F\ x_1...x_k \to e \in \mathcal{R}$, let $t' = C[e_{[\forall i\ x_i \mapsto t_i]}]$.

First, given a ground type context $C[\bullet^o]$, we can define the associated ground type context $\|C\|[\bullet^o]$ by adding to Definition 1 the fact $\|\ \bullet^o\ \| = \bullet^o$. Hence we can say that $\|t\| = \|C\|[\overline{F}\ \overline{t_1}...\overline{t_k}\ \Delta]$.

We see that $\|t\| \to_{\overline{G}} \|C\|[\overline{e}_{[\forall i\ x_i \mapsto \overline{t_i}]}\ \Delta]$. Claim A shows that $\overline{e}_{[\forall i\ x_i \mapsto \overline{t_i}]}\ \Delta = \overline{e_{[\forall i\ x_i \mapsto t_i]}}\Delta \to^*_{\overline{G}} \|e_{[\forall i\ x_i \mapsto t_i]}\|$, hence $\|t\| \to_{\overline{G}} \|C\|[\overline{e}_{[\forall i\ x_i \mapsto \overline{t_i}]}\ \Delta] \to^*_{\overline{G}} \|C\|[\|e_{[\forall i\ x_i \mapsto t_i]}\|] = \|t'\|$.

Claim (♠). Given a term $t' \in \mathrm{Acc}_{\overline{G}}$ there exists a term $t \in \mathrm{Acc}_G$ such that $t' \to_{\overline{G}} \|t\|$. **This claim implies property** (2).

Proof (Claim (♠)). We will divide the relation $\to_{\overline{G}}$ in two relations : $\to_{\overline{G}} = \to_\Sigma \uplus \to_\mathcal{N}$ depending of the head symbol of the redex we're rewritting. Let $t \to_{\overline{G}} t'$ if the rewrite rule applied is $r_{\overline{a}}$ for some $a \in \Sigma$ then $t \to_\Sigma t'$, if the rewrite rule is $r_{\overline{F}}$ with $F \in \mathcal{N}$, then $t \to_\mathcal{N} t'$.

The proof is in four steps:

1. Given a term $t \in \mathrm{Acc}_{\overline{G}}$, there is only a finite number of derivation $t \to^*_\Sigma t'$, furthermore, if $t \to^*_\Sigma t_1$ and $t \to^*_\Sigma t_2$ such that there is no t' such that $t_1 \to^*_\Sigma t'$ or $t_2 \to^*_\Sigma t'$, then $t_1 = t_2$. We name this unique term t^Σ, and we notice that if $t \to^*_\Sigma t'$ then $t' \to^*_\Sigma t^\Sigma$. Basically this step comes from the fact that the relation $\to_\Sigma$ strictly decrease the number of occurrences of terms headed by some $\overline{a}$ with $a \in \Sigma$.
2. Let $t = C[\overline{F}\ t_1...t_{\mathrm{arity}(\overline{F})}]$, then $t^\Sigma = C^\Sigma[\overline{F}\ t_1...t_{\mathrm{arity}(\overline{F})}]$, C^Σ being defined inductively: if $C[\bullet] = \bullet$ then $C^\Sigma[\bullet] = \bullet$, if $C[\bullet] = a\ t_1...C'[\bullet]...t_k$, then $C^\Sigma[\bullet] = a\ t_1^\Sigma...C'^\Sigma[\bullet]...t_k^\Sigma$ (Claim A shows that these are the only possibilities). This step is shown by induction.
3. If $t \to_\mathcal{N} t'$ i.e. $t = C[\overline{F}\ t_1...t_k]$ and $t' = C[\overline{e}_{[\forall i\ x_i \mapsto \overline{t_i}]}\ \Delta]$ with the appropriate $\overline{e}$, let $t'' = C^\Sigma[\overline{e}_{[\forall i\ x_i \mapsto \overline{t_i}]}\ \Delta]$, then $t'^\Sigma = t''^\Sigma$.
4. Finally we prove by induction that for all t', there exists $t \in \mathrm{Acc}_G$ such that $t'^\Sigma = \|t\|$, which proves the claim.

B The Type System Detailed

We give here a complete proof of Theorem 3 and Proposition 1. We first recall the type system, and the definition of $\vdash (G, t) \triangleright \sigma$.

$$\frac{\Theta \vdash t \triangleright \theta_1 \quad ... \quad \Theta \vdash t \triangleright \theta_n}{\Theta \vdash t \triangleright \bigwedge\{\theta_1, ..., \theta_n\}}(Set)$$

$$\frac{}{\Theta, \alpha \triangleright \bigwedge\{\theta_1, ..., \theta_n\} \vdash \alpha \triangleright \theta_i}(At) \qquad \textit{(for all i)}$$

$$\frac{}{\Theta \vdash a \triangleright \sigma_1 \to ... \to \sigma_{i \leq arity(a)} \to q_\infty}(\Sigma)(\textit{ for } a\ a \in \Sigma \textit{ and } \exists j\ \sigma_j = q_\infty\ )$$

$$\frac{\Theta \vdash t_1 \triangleright \sigma \to \theta \quad \Theta \vdash t_2 \triangleright \sigma}{\Theta \vdash t_1\ t_2 \triangleright \theta}(App)$$

$$\frac{}{\Theta \vdash t \triangleright q_\infty \to q_\infty}(q_\infty \to q_\infty)\ (\mathit{if}\ t : \tau_1 \to \tau_2)$$

$$\frac{\Theta \vdash t_1 \triangleright q_\infty}{\Theta \vdash t_1\ t_2 \triangleright q_\infty}(q_\infty)$$

Remark that:

- Using rule (Set) one can always prove, for any term t, $\Theta \vdash t \triangleright \bigwedge\{\}$.
- $\Theta \vdash t \triangleright \bigwedge\{\theta_1, ..., \theta_k\}$ if and only if , for all i, $\Theta \vdash t \triangleright \theta_i$.
- There is no rules that directly involve $q_\perp$, but that does not mean that no term matches $q_\perp$, since it can appears in Θ. Rules like (At) or (App) may be use to state that a term matches $q_\perp$.

We say that (G, t) matches the conjunctive mapping σ written $\vdash (G, t) \triangleright \sigma$ if there exists an environment Θ, called a witness environment of $\vdash (G, t) \triangleright \sigma$, which verifies the following properties:

1. $Dom(\Theta) = \mathcal{N}$,
2. $\forall F : \tau \in \mathcal{N}\ \Theta(F) :: \tau$,
3. if $F\ x_1...x_k \to e \in \mathcal{R}$ and $\Theta \vdash F \triangleright \sigma_1 \to ... \to \sigma_{i \leq k} \to q$ then either there exists j such that $q_\infty \in \sigma_j$, or $i = k$ and $\Theta, x_1 \triangleright \sigma_1, ..., x_k \triangleright \sigma_k \vdash e \triangleright q$,
4. $\Theta \vdash t \triangleright \sigma$.

Lemma 2 (Isolated Non Terminals). *Given a non terminal F that has not ground type. Then if Θ verifies properties* 1 *to* 3*, one cannot prove $\Theta \vdash F \triangleright q_\infty$.*

Proof (Proof of lemma 2). The proof comes from the fact that Θ verifies property 3. Assume $\Theta \vdash F \triangleright \sigma_1 \to ... \to \sigma_i \to q_\infty$. Property 3 states that if $i < \text{arity}(F)$ then there exists $j \leq i$ such that $q_\infty \in \sigma_j$ in particular $i \neq 0$. If $i = \text{arity}(F)$ then by hypothesis, $i \neq 0$. Then, one cannot prove $\Theta \vdash F \triangleright q_\infty$.

Lemma 3 (Non fully-applied terminals). *Let Θ be an environment that verifies properties* 1 *to* 3*, let F be a non terminal that has not ground type and let $t = F\ t_1...t_i$ with $i <$ arity(F). If $\Theta \vdash F\ t_1...t_i \triangleright q_\infty$ then there exists $j \leq i$ such that $\Theta \vdash t_i \triangleright q_\infty$.*

Proof (Proof of Lemma 3). We prove by induction on l the following more general result: if $\Theta \vdash F\ t_1...t_l \triangleright \sigma_{l+1} \to ... \to \sigma_i \to q_\infty$ with $F \in \mathcal{N}$ and $i < \text{arity}(F)$, then there exists $j \leq i$ such that $\Theta \vdash t_j \triangleright q_\infty$ if $j \leq l$ or $q_\infty \in \sigma_j$ if $j > l$.

If $l = 1$, then the rule we used to prove $\Theta \vdash F\ t_1 \triangleright \sigma_2 \to ... \to \sigma_i \to q_\infty$ could not be (q_∞) since one cannot prove $\Theta \vdash F \triangleright q_\infty$. If the rule we used were $(q_\infty \to q_\infty)$, then $q_\infty \in \sigma_2$. If it is rule (App) then $\Theta \vdash F \triangleright \sigma_1 \to \sigma_2 \to ... \to \sigma_i \to q_\infty$ and $\Theta \vdash t_1 \triangleright \sigma_1$, and since $i < \text{arity}(F)$, property 3 states that there exists $j < l$ such that $\Theta \vdash t_1 \triangleright q_\infty$ if $j = 1$, $q_\infty \in \sigma_j$ elseway. These are the only rules we could have applied.

If $l > 1$. If we applied rule (q_∞) then $\Theta \vdash F\ t_1...t_{l-1} \triangleright q_\infty$ by induction hypothesis there exists $j \leq l - 1$ such that $\Theta \vdash t_j \triangleright q_\infty$. If we applied rule

$(q_\infty \to q_\infty)$, then $q_\infty \in \sigma_{l+1}$. If we applied rule (App) then $\Theta \vdash F\ t_1...t_{l-1} \triangleright \sigma_l \to \sigma_{l+1} \to ... \to \sigma_i \to q_\infty$ and $\Theta \vdash t_l \triangleright \sigma_1$ then by induction hypothesis there exists $j \leq i$ such that either $\Theta \vdash t_j \triangleright q_\infty$ if $j < l$ or $q_\infty \in \sigma_j$ if $j \geq l$. If $j \leq l$ then either $j < l$ and then $\Theta \vdash t_j \triangleright q_\infty$, or $j = l$ and $q_\infty \in \sigma_j$ hence $\Theta \vdash t_j \triangleright q_\infty$, if $j > l$ then $q_\infty \in \sigma_j$ if $j \geq l$.

Lemma 4 (Redexes and q_∞).
(1) If Θ verifies properties 1 to 3, then given a term t, if $\Theta \vdash t \triangleright q_\infty$ then either t contains a redex r such that $\Theta \vdash r \triangleright q_\infty$. In particular, if t does not contains any redex, then one cannot prove $\Theta \vdash t \triangleright q_\infty$.

(2) If Θ verifies properties 1 to 3, then given a term t, if t contains a redex r such that $\Theta \vdash r \triangleright q_\infty$ then one can prove $\Theta \vdash t \triangleright q_\infty$.

Proof (Lemma 4). We prove (1) by induction on t. Assume that $\Theta \vdash t \triangleright q_\infty$.

If $t : o$ then either $t = F\ t_1...t_k$ in which case t is the redex r, or $t = a\ t_1...t_k$ then the ony rule we could have applied to prove $\Theta \vdash t \triangleright q_\infty$ is (Σ), then there exists $t_i : o$ such that $\Theta \vdash t_i \triangleright q_\infty$, and the result comes by induction.

If $t : \tau$ with $\tau \neq o$. We could not have $t = F \in \mathcal{N}$ since one cannot prove $\Theta \vdash F \triangleright q_\infty$ if F has not ground type. If $t = a\ t_1...t_i$ then again there exists $t_i : o$ such that $\Theta \vdash t_i \triangleright q_\infty$, and the result comes by induction. If $t = F\ t_1...t_i$, F has not ground type, and $i < \text{arity}(F)$ since t has not ground type. Then Lemma 3 states that there exists t_j such that $\Theta \vdash t_j \triangleright q_\infty$ and the result comes by induction.

To prove (2), assume that there is a redex r such that $\Theta \vdash r \triangleright q_\infty$ and $t = C[r]$. We prove the result by induction on $C[\bullet]$. If $C[\bullet] = \bullet$, then $t = r$ therefore $\Theta \vdash r \triangleright q_\infty$. Assume $t = t_1\ t_2$ with $t_1 = C'[r]$ or $t_2 = C'[r]$. If $t_1 = C'[r]$ then by induction hypothesis, one can prove $\Theta \vdash t_1 \triangleright q_\infty$ and then, using rule $q_\infty\ I$, $\Theta \vdash t_1\ t_2 \triangleright q_\infty$. If $t_2 = C'[r]$, by induction hypothesis, one can prove $\Theta \vdash t_2 \triangleright q_\infty$, using rule $(q_\infty \to q_\infty)$, we have $\Theta \vdash t_1 \triangleright q_\infty \to q_\infty$ and then, rule (App) gives us $\Theta \vdash t_1\ t_2 \triangleright q_\infty$.

Lemma 5 (Ground type terms and $q_\perp$). *if Θ verifies properties 1 to 3, then if $t : \tau$ and $\Theta \vdash t \triangleright \sigma$, $\sigma :: \tau$. In particular, if $\Theta \vdash t \triangleright q_\perp$, then $t : o$.*

Proof (Lemma 5). We can assume, without loss of generality that $\sigma = \{\theta\}$ for some atomic mapping θ. We prove this by induction on the structure of t.

If $t = \alpha$ with $\alpha \in \Sigma \uplus \mathcal{N}$, then the only rules we can apply are (At), (Σ) and $(q_\infty \to q_\infty)$ and they all satisfy the property.

If $t = t_1\ t_2$ with $t_1 : \tau_2 \to \tau$ and $t_2 : \tau_2$, then the rules we can apply are either (q_∞) or (App). If it is (q_∞) then we have proven $\Theta \vdash t \triangleright q_\infty$ and $q_\infty :: \tau$. If it is (App) it means that we have proven $\Theta \vdash t_1 \triangleright \sigma' \to \theta$ and $\Theta \vdash t_2 \triangleright \sigma'$, and by induction hypothesis, $\sigma' :: \tau_2$ and $\theta :: \tau$.

Theorem 6 (Soundness). *Let G be an HORS, and t be term (of any type), if $\vdash (G, t) \triangleright q_\infty$ (resp. $\vdash (G, t) \triangleright q_\perp$) then $\mathcal{P}_\infty(t)$ (resp. $\mathcal{P}_\perp(t)$) holds.*

Proof (Theorem 6).

Lemma 6 (Type Preservation). *Let $t : \tau$ be a term. If $\vdash (G,t) \triangleright \sigma$ and $t \rightarrow_{IO} t'$ then $\vdash (G,t') \triangleright \sigma$.*

Proof (Lemma 6). Assume that $\vdash (G,t) \triangleright \sigma$ and $t \rightarrow_{IO} t'$. Let Θ be a witness environment of $\vdash (G,t) \triangleright \sigma$, we will prove that it is also a witness environment of $\vdash (G,t') \triangleright \sigma$ (we only have to check that $\Theta \vdash t' \triangleright \sigma$).

We know that $t = C[F\ s_1...s_k]$ and $t' = C[e_{[\forall i\ x_i \mapsto s_i]}]$ for some context $C[\bullet : o] : \tau$. We proceed by induction on $C[\bullet]$.

If $C[\bullet] = \bullet$, we can assume without loss of generality that $\sigma = q \in Q$. We look at the proof of $\vdash (G,t') \triangleright \sigma$ and remark that either (1) the proof contains $\Theta \vdash F \triangleright \sigma_1 \rightarrow ... \rightarrow \sigma_k \rightarrow \sigma$ and $\Theta \vdash s_i \triangleright \sigma_i$ for all i and the last steps are using the rule (*App*), or (2) the proving tree contains $\Theta \vdash F\ s_1...s_i \triangleright q_\infty \rightarrow q_\infty$ and $\Theta \vdash s_{i+1} \triangleright q_\infty$, the last steps are using the rule (*App*) once and then only rule $(q_\infty I)$. The former case is impossible: since $t \rightarrow_{IO} t'$ is an IO derivation there's no redex in s_i for all i, hence Lemma 4 shows that one cannot have $\Theta \vdash s_{i+1} \triangleright q_\infty$.

Hence the proving tree contains $\Theta \vdash F \triangleright \sigma_1 \rightarrow ... \rightarrow \sigma_k \rightarrow \sigma$ and $\Theta \vdash s_i \triangleright \sigma_i$ for all i, then $\Theta, x_1 \triangleright \sigma_1, ..., x_k \triangleright \sigma_k \vdash e \triangleright q$, and if we replace all statements of $x_i \triangleright \sigma_i$ by the proof of $\Theta \vdash s_i \triangleright \sigma_i$, we obtain a proof of $\Theta \vdash e_{[\forall i\ x_i \mapsto s_i]} \triangleright \sigma$.

Now we prove the induction step. Assume that $C = C'[\bullet]\ t_2$ or $C = t_1\ C'[\bullet]$, then$t = t_1\ t_2$ with either $t_1 = C'[F\ s_1...s_k]$ or $t_2 = C'[F\ s_1...s_k]$. Then $t' = t'_1\ t'_2$ with respectively, either $t'_1 = C'[e_{[\forall i\ x_i \mapsto s_i]}]$ and $t'_2 = t_2$, or $t'_1 = t_1$ and $t'_2 = C'[e_{[\forall i\ x_i \mapsto s_i]}]$. Either way, by induction hypothesis, if $\Theta \vdash t_1 \triangleright \sigma_1$ (resp. $\Theta \vdash t_2 \triangleright \sigma_2$), then $\Theta \vdash t'_1 \triangleright \sigma_1$ (resp. $\Theta \vdash t'_2 \triangleright \sigma_2$). Assume we have proven $\Theta \vdash t \triangleright \sigma$. In order to do it, either we have use rule $(q_\infty\ I)$ or rule (*App*), either way we could use the same rule to prove $\Theta \vdash t' \triangleright \sigma$.

We extend in an intuitive way the properties $\mathcal{P}_\perp$ and $\mathcal{P}_\infty$ to trees: if t is a tree then $\mathcal{P}_\perp(t)$ ="$t = \perp$" and $\mathcal{P}_\infty(t)$ ="t is either infinite or contains $\perp$.".

Lemma 7 (Weak Soundness). *Given a term $t : o$, (1) if $\vdash (G,t) \triangleright q_\perp$, then $\mathcal{P}_\perp(t^\perp)$ holds, (2) if $\vdash (G,t) \triangleright q_\infty$, then $\mathcal{P}_\infty(t^\perp)$ holds.*

Proof (Lemma 7). We can use Lemma 4 to prove (2): if $\vdash (G,t) \triangleright q_\infty$ then t contains a redex, hence $t^\perp$ contains $\perp$, therefore $\mathcal{P}_\infty(t^\perp)$ holds.

We prove (1) by induction on the structure of $t^\perp$. If $t^\perp = \perp$ then $P_\perp(t^\perp)$ is true hence (1) holds. If $t^\perp = a$ with $a \in \Sigma$, then $t = a$ and there is no rule that we can apply to state $\vdash (G,a) \triangleright q_\perp$, hence (1) and (2) holds. If $t^\perp = a\ t'_1...t'_k$ with $k > 0$, then $t = a\ t_1...t_k$ with $a \in \Sigma$ and $t_i^\perp = t'_i$ for all i. For all environment Θ, we show by induction that for all i, if $\Theta \vdash a\ t_1...t_i \triangleright \sigma'$ then $\sigma' = \sigma_1 \rightarrow ... \rightarrow \sigma_l \rightarrow q_\infty$: The term a can only be judge by the rule (Σ) hence it is true if $i = 0$, the term $(a\ t_1...t_i)\ t_{i+1}$ can be judge by rules (q_∞), $(q_\infty \rightarrow q_\infty)$ and (*App*) and by induction hypothesis, in all three cases, we get $\Theta \vdash a\ t_1...t_i \triangleright \sigma_1 \rightarrow ... \rightarrow \sigma_l \rightarrow q_\infty$ for some l. In particular, we don't have $\vdash (G,t) \triangleright q_\perp$, hence (1) holds.

Using Lemma 1 and 2, in order to prove Therorem 6 we can assume that $t : o$. We prove it by contradiction. Assume that $\vdash (G,t) \triangleright q_\infty$ but $\mathcal{P}_\infty(t)$ doesn't hold. Then it means that $\|G_t\|$ is finite and contains only terminals. Since it's finite, there exists a finite IO derivation from t that leads to $\|G_t\|$: $t \rightarrow^*_{IO} \|G_t\|$, hence using Lemmas 6 and 7 we can prove $\mathcal{P}_\infty((\|G_t\|)^\perp)$, but since $\|G_t\|$ is a tree, $\|G_t\| = (\|G_t\|)^\perp$, hence $\|G_t\|$ is infinite or contains $\perp$ which raises a contradiction.

We treat the case $\vdash (G,t) \triangleright q_\perp$ the same way: Assume that $\vdash (G,t) \triangleright q_\perp$ but $\mathcal{P}_\perp(t)$ doesn't hold. Then it means that $\|G_t\|$ contains some terminals. Then there exists a finite IO derivation from t that leads to a term t' such that $t'^\perp \neq \perp$: $t \rightarrow^*_{IO} t'$, hence using Lemmas 6 and 7 we can prove $\mathcal{P}_\perp((t')^\perp)$ which is false.

Theorem 7 (Completeness). *Let G be an HORS, if $\mathcal{P}_\infty(t)$ (resp. $\mathcal{P}_\perp(t)$) holds then $\vdash (G,t) \triangleright q_\infty$ (resp. $\vdash (G,t) \triangleright q_\perp$).*

Proof (Proof of Theorem 7).

Using Lemma 4 we can assume without loss of generality that t has ground type.

We recall the properties that an environment Θ has to satisfy in order to be a witness of $\vdash (G,t) \triangleright \sigma$.

1. $Dom(\Theta) = \mathcal{N}$,
2. $\forall F : \tau \in \mathcal{N}\ \Theta(F) :: \tau$,
3. if "$F\ x_1...x_k \rightarrow e$"$\in \mathcal{R}$ and $\Theta \vdash F \triangleright \sigma_1 \rightarrow ... \rightarrow \sigma_{i\leq k} \rightarrow q$ then either there exists j such that $q_\infty \in \sigma_j$, or $i = k$ and $\Theta, x_1 \triangleright \sigma_1, ..., x_k \triangleright \sigma_k \vdash e \triangleright q$,
4. $\Theta \vdash t \triangleright \sigma$.

Let $\mathcal{E}$ be the set of environment that matches properties 1 and 2. Let $\mathcal{F} : \mathcal{E} \rightarrow \mathcal{E}$ be a mapping such that for all $F : \tau_1 \rightarrow ... \rightarrow \tau_k \rightarrow o \in \mathcal{N}$, if $F\ x_1...x_k \rightarrow e \in \mathcal{R}$ then,

$$\begin{aligned}\mathcal{F}(\Theta)(F) = \{\sigma_1 \rightarrow ... \rightarrow \sigma_k \rightarrow q \mid q \in Q \wedge \forall i\ \sigma_i :: \tau_i \wedge \Theta, x_1 \triangleright \sigma_1, ..., x_k \triangleright \sigma_k \vdash e : q\}\\ \cup \{\sigma_1 \rightarrow ... \rightarrow \sigma_{i\leq k} \rightarrow q_\infty \mid\ \wedge \forall i\ \sigma_i :: \tau_i \wedge \exists j\ q_\infty \in \sigma_j\}\\ \cup \{\sigma_1 \rightarrow ... \rightarrow \sigma_k \rightarrow q_\perp \mid \forall i\ \sigma_i :: \tau_i \wedge \exists j\ q_\infty \in \sigma_j\}.\end{aligned}$$

Let $\Theta_0 \in \mathcal{E}$ be the environment such that, for all $F : \tau = \tau_1 \rightarrow ... \rightarrow \tau_k \rightarrow o \in \mathcal{N}$, $\Theta_0(F)$ is defined and contains all atomic mappings $\theta ::_a \tau$. Notice that:

$$\Theta_0(F) = \{\sigma_1 \rightarrow ... \rightarrow \sigma_k \rightarrow q \mid q \in Q \wedge \forall i\ \sigma_i :: \tau_i\} \cup \{\sigma_1 \rightarrow ... \rightarrow \sigma_{i<k} \rightarrow q_\infty \mid \forall j\ \sigma_j :: \tau_j\}.$$

Lemma 8 (Universal Witness). *There exists $m \in \mathbb{N}$ such that the judgment $\vdash (G,t) \triangleright \sigma$ holds if and only if $\mathcal{F}^m(\Theta_0) \vdash t \triangleright \sigma$ (This is Proposition 1 with $\Theta^\star = \mathcal{F}^m(\Theta_0)$).*

Proof (Proof of Lemma 8). We define the partial order $\sqsubseteq$ on $\mathcal{E}$ such that $\Theta_1 \sqsubseteq \Theta_2$ if and only if, for all $F \in \mathcal{N}$, $\Theta_1(F) \subseteq \Theta_2(F)$. Note that if $\Theta_1 \sqsubseteq \Theta_2$ and $\Theta_1 \vdash t \triangleright \sigma$ then $\Theta_2 \vdash t \triangleright \sigma$. $\Theta_0(F)$ contains all atomic mappings $\theta ::_a \tau$, hence Θ_0 is a maximum of $\mathcal{E}$ with respect to $\sqsubseteq$. Note that the mapping $\mathcal{F}$ is monotonic with respect to $\sqsubseteq$ (i.e. if $\Theta \sqsubseteq \Theta'$ then $\mathcal{F}(\Theta) \sqsubseteq \mathcal{F}(\Theta')$). Given $\Theta \in \mathcal{E}$, we say that Θ is a post-fixpoint of $\mathcal{F}$ if and only if $\Theta \sqsubseteq \mathcal{F}(\Theta)$. Remark that being a post-fixpoint of $\mathcal{F}$ is the same as verifying property 3.

Since Θ_0 is a maximum of $\mathcal{E}$, and $\mathcal{F}(\Theta_0) \in \mathcal{E}$, then $\Theta_0 \sqsupseteq \mathcal{F}(\Theta_0)$, therefore, since $\mathcal{F}$ is monotonic, $\Theta_0 \sqsupseteq \mathcal{F}(\Theta_0) \sqsupseteq \mathcal{F}^2(\Theta_0) \sqsupseteq ....$ Because $\mathcal{E}$ is finite, there exists m such that $\mathcal{F}^m(\Theta_0) = \mathcal{F}^{m+1}(\Theta_0)$, in particular $\mathcal{F}^m(\Theta_0)$ is a post-fixpoint of $\mathcal{F}$, hence it verifies properties 1, 2, and 3.

Take a witness Θ of $\vdash (G,t) \triangleright \sigma$. Θ is a post-fixpoint of $\mathcal{F}$, and since $\mathcal{F}^m(\Theta_0)$ is the greatest post-fixpoint, $\mathcal{F}^m(\Theta_0) \sqsupseteq \Theta$, hence $\mathcal{F}^m(\Theta_0) \vdash t \triangleright \sigma$, thus $\mathcal{F}^m(\Theta_0)$ is a witness of $\vdash (G,t) \triangleright \sigma$.

Let $G^{(m)} = \langle \mathcal{V}, \Sigma, \mathcal{N}^{(m)} \uplus \{Void : o\}, \mathcal{R}^{(m)}, I \rangle$ be the scheme such that $\mathcal{N}^{(m)} = \bigcup_{0 \leq i \leq m} \{F_i \mid F \in \mathcal{N}\}$. For all $F\ x_1...x_k \to e \in \mathcal{R}$, $\mathcal{R}^{(m)}$ contains the following rewrite rules:

$$F_i\ x_1...x_k \to e_{[\forall H \in \mathcal{N}\ H \mapsto H_{i-1}]} \quad for\ i > 0$$

$$F_0\ x_1...x_k \to e_{[\forall H \in \mathcal{N}\ H \mapsto H_0]}$$

$$F_0\ x_1...x_k \to Void$$

$$Void \to Void$$

Notice that $Void$ here is a non-terminal of order 0 that produce itself. Hence applying its rewrite rule to a term would produce the same term. In the following we forbid this rule to be applied. $G^{(m)}$ with this restriction is said to be recursion free, i.e. the graph whose vertices are the non terminals and where there is an edge from F to G if and only if there exist an allowed rewrite rule $F\ x_1...x_k \to e$ such that e contains an occurence of G, has no loop. Such non-recursive schemes are known to be strongly normalizing, i.e. for any term t all derivations using only allowed rewrite rules are finite. In particular, there exists a finite IO derivation $t \to_{IO}^* t'$ such that $(t')^{\perp} = \|t\|_{IO}$.

We define the environment $\Theta^{(m)}$ on $\mathcal{N}^{(m)} \uplus \{Void\}$: for all $F \in \mathcal{N}$, for all $i \leq j$, $\Theta^{(m)}(F_i) = \mathcal{F}^i(\Theta_0)(F)$ and $\Theta^{(m)}(Void) = \bigwedge\{q_\infty, q_\perp\}$.

Lemma 9. *Given a two terms $t, t' \in \mathcal{T}(\Sigma \uplus \mathcal{N}^{(m)})$ such that $t \to_{IO} t'$ is allowed in $G^{(m)}$. If $\Theta^{(m)} \vdash t' \triangleright \sigma$, then $\Theta^{(m)} \vdash t \triangleright q$.*

Proof (Lemma 9). We proceed by induction on the structure of t. We prove the initial case: $t = F_l\ s_1...s_k$ and $t' = e_{[\forall i\ x_i \mapsto s_i]}$, with $F_l\ x_1...x_k \to e \in \mathcal{R}$. We assume without loss of generality that σ is atomic, hence $\sigma = q \in Q$. Let σ_i be the union of all mappings assigned to s_i in the proof of $\Theta^{(m)} \vdash e_{[\forall i\ x_i \mapsto s_i]} \triangleright q$. Then we have $\Theta^{(m)}, x_1 \triangleright \sigma_1, ..., x_k \triangleright \sigma_k \vdash e \triangleright q$. Let $\Theta' = \Theta^{(m)}, F_l \triangleright \sigma_1 \to ... \to \sigma_k \to q$. Since $\Theta' \vdash F_l \triangleright \sigma_1 \to ... \to \sigma_k \to q$, and $\Theta' \vdash s_i \triangleright \sigma_i$ (indeed, $\Theta^{(m)} \subseteq \Theta'$), we can prove $\Theta' \vdash t \triangleright q$. If $l = 0$, by definition, $F \triangleright \sigma_1 \to ... \to \sigma_k \to q \in \Theta_0(F)$, and since $\Theta^{(m)}(F_0) = \Theta_0(F)$, we have $F \triangleright \sigma_1 \to ... \to \sigma_k \to q \in \Theta^{(m)}(F_0)$, hence $\Theta' = \Theta^{(m)}$. If $l > 0$, e only contains terminals of the form G_{l-1}, then we can transform the proof of $\Theta^{(m)}, x_1 \triangleright \sigma_1, ..., x_k \triangleright \sigma_k \vdash e \triangleright q$ to obtain a proof of $\mathcal{F}^{l-1}(\Theta_0), x_1 \triangleright \sigma_1, ..., x_k \triangleright \sigma_k \vdash e_{[\forall G \in \mathcal{N}\ G_{l-1} \mapsto G]} \triangleright q$. Then by definition of $\mathcal{F}$, $F \triangleright \sigma_1 \to ... \to \sigma_k \to q \in \mathcal{F}^l(\Theta_0)(F)$, and since $\Theta^{(m)}(F_l) = \mathcal{F}^l(\Theta_0)(F)$,we have $\sigma_1 \to ... \to \sigma_k \to q \in \Theta^{(m)}(F_l)$, hence $\Theta' = \Theta^{(m)}$. Thus $\Theta^{(m)} \vdash t \triangleright q$.

For the induction step, We assume without loss of generality that $\sigma = \bigwedge\{\theta\}$. Assume that $C = C'[\bullet]\ t_2$ or $C = t_1\ C'[\bullet]$, then $t = t_1\ t_2$ with either $t_1 = C'[F\ s_1...s_k]$ or $t_2 = C'[F\ s_1...s_k]$. Then $t' = t'_1\ t'_2$ with respectively, either $t'_1 = C'[e_{[\forall i\ x_i \mapsto s_i]}]$ and $t'_2 = t_2$, or $t'_1 = t_1$ and $t'_2 = C'[e_{[\forall i\ x_i \mapsto s_i]}]$. Either way, by induction hypothesis, if $\Theta^{(m)} \vdash t'_1 \triangleright \sigma_1$ (resp. $\Theta^{(m)} \vdash t'_2 \triangleright \sigma_2$), then $\Theta^{(m)} \vdash t_1 \triangleright \sigma_1$ (resp. $\Theta^{(m)} \vdash t_2 \triangleright \sigma_2$). Assume we have proven $\Theta^{(m)} \vdash t' \triangleright \sigma$. In order to do it, either we have used rule $(q_\infty\ I)$ or rule (App), either way we could use the same rule to prove $\Theta^{(m)} \vdash t \triangleright \sigma$.

Lemma 10 (Terms that contains $Void$). *Given a term t that contains the non terminal $Void$, one can prove $\Theta^{(m)} \vdash t \triangleright q_\infty$.*

Proof (Lemma 10). We prove the result by induction on the structure of t : if $t = Void$ then we use rule (At) , if $t = G_j\ t_1...t_i$ with $G_j \in \mathcal{N}^{(m)}$ and t_ℓ contains $Void$ for some $\ell \leq i$, then by induction hypothesis, $\Theta^{(m)} \vdash t_\ell \triangleright q_\infty$. Using rule (Set) we have, for all $\ell' \neq \ell$, $\Theta^{(m)} \vdash t_{\ell'} \triangleright \bigwedge\{\}$. We have by construction $\Theta^{(m)} \vdash a \triangleright \sigma_1 \to ... \to \sigma_k \to q_\infty$ for $\sigma_j = q_\infty$ and $\sigma_i = \bigwedge\{\}$ for all $i \neq j$. Then, if we apply k times rule (App) we can prove $\Theta^{(m)} \vdash a\ t_1...t_k \triangleright q_\infty$. By definition, if $\sigma_{\ell' \neq \ell} = \bigwedge\{\}$ and $\sigma_\ell = q_\infty$, we have $\sigma_1 \to ... \to \sigma_i \to q_\infty \in \Theta^{(m)}(G_j)$. Hence one can prove $\Theta^{(m)} \vdash t_i \triangleright q_\infty$.

if $t = G_j\ t_1...t_i$ with $a \in \Sigma$ and t_ℓ contains $Void$ for some $\ell \leq i$ the proof is similar except we use rule (Σ) to prove $\Theta^{(m)} \vdash a \triangleright \sigma_1 \to ... \to \sigma_i \to q_\infty$ with $\sigma_{\ell' \neq \ell} = \bigwedge\{\}$ and $\sigma_\ell = q_\infty$.

Lemma 11 (Weak Completeness). *Given a term $t : o \in \mathcal{T}(\Sigma \uplus \mathcal{N}^{(m)})$, if $\mathcal{P}_\perp(t^\perp)$ (resp. $\mathcal{P}_\infty(t^\perp)$) holds and if there exists no IO allowed rewrite rule we can apply in t, then $\Theta^{(m)} \vdash t \triangleright q_\perp$ (resp. $\emptyset \vdash t^\perp \triangleright q_\infty$).*

Proof (Lemma 11).

We prove both results simultaneously by induction on the structure of t. If $t = Void$ then one can directly prove both result using rule (At). We know that $t \neq a$ with $a \in \Sigma$ since $\mathcal{P}_\perp(a)$ (resp. $\mathcal{P}_\infty(a)$) does not hold. If $t = a\ t_1...t_k$, we know that $\mathcal{P}_\perp(t^\perp)$ doesn't hold, assume that $\mathcal{P}_\infty(t^\perp)$ holds. Since $t^\perp$ contains $\perp$, there exists j such that $t_j^\perp$ contains $\perp$, i.e. $\mathcal{P}_\infty(t_j^\perp)$. Furthermore there exists no allowed rewrite rule we can apply in t_j, elseway we could apply it in t. Therefore, by induction hypothesis, $\Theta^{(m)} \vdash t_j \triangleright q_\infty$. Using rule (Set) we have, for all $i \neq j$, $\Theta^{(m)} \vdash t_i \triangleright \bigwedge\{\}$. Rule (Σ) gives $\Theta^{(m)} \vdash a \triangleright \sigma_1 \to ... \to \sigma_k \to q_\infty$ for $\sigma_j = q_\infty$ and $\sigma_i = \bigwedge\{\}$ for all $i \neq j$. Then, if we apply k times rule (App) we can prove $\Theta^{(m)} \vdash a\ t_1...t_k \triangleright q_\infty$.

Assume now that $t = F_l\ t_1...t_k$ with $F_l \in \mathcal{N}^{(m)}$. Since there exists no *IO* allowed rewrite rule we can apply in t it means that there exists i such that t_i contains a redex, but this redex can't be applied, in other words, t_i contains $Void$. Using Lemma 10 we have $\Theta^{(m)} \vdash t_i \triangleright q_\infty$. By definition, if $\sigma_{j \neq i} = \bigwedge\{\}$ and $\sigma_i = q_\infty$, we have $\sigma_1 \to ... \to \sigma_k \to q \in \Theta^{(m)}(F_l)$ for $q \in Q$. Hence one can prove $\Theta^{(m)} \vdash t \triangleright q$.

Now, we can prove the Theorem. Given a term $t : o \in \mathcal{T}(\Sigma \uplus \mathcal{N})$, assume that $\mathcal{P}_\perp(t)$ (resp. $\mathcal{P}_\infty(t)$) holds. We define the term $t^{(m)} : o = t_{[\forall F \in \mathcal{N}\ F \mapsto F_i]} \in \mathcal{T}(\Sigma \uplus$

$\mathcal{N}^{(m)}$). Notice that $\|G^{(m)}_{t^{(m)}}\|_{IO}$ is obtained by turning some subtrees of $\|G_t\|_{IO}$ into $\perp$. Hence, $\mathcal{P}_\perp(t^{(m)})$ (resp. $\mathcal{P}_\infty(t^{(m)})$) holds. Let $t' : o \in \mathcal{T}(\Sigma \uplus \mathcal{N}^{(m)} \uplus \{\perp\})$ such that $t^{(m)} \rightarrow^*_{IO} t'$ and $(t')^\perp = \|G^{(m)}_{t^{(m)}}\|_{IO}$ (we have seen previously that such t' exists). Lemma 11 states that $\Theta^{(m)} \vdash t' \triangleright q_\perp$ (resp. $\Theta^{(m)} \vdash t' \triangleright q_\infty$), then, Lemma 9 shows that $\Theta^{(m)} \vdash t^{(m)} \triangleright q_\perp$ (resp. $\Theta^{(m)} \vdash t^{(m)} \triangleright q_\infty$). Since non terminals in $t^{(m)}$ have the form F_m , if we restrict the domain of $\Theta^{(m)}$ only to $\{F_m \mid F \in \mathcal{N}\}$ the proof still holds, furthermore in this proof, if we remove all "m" subscripts, we get $\mathcal{F}^m(\Theta_0) \vdash t \triangleright q_\perp$ (resp. $\mathcal{F}^m(\Theta_0) \vdash t \triangleright q_\infty$). Lemma 8 allows us to conclude: $\vdash (G, t) \triangleright q_\perp$ (resp. $\vdash (G, t) \triangleright q_\infty$).

C Selfcorrecting Scheme

Proof (Theorem 4).

Lemma 12 (Equality of Trees). *Let $t : o \in \mathcal{T}(\mathcal{V} \uplus \mathcal{N})$ be a term, then $t^\perp = (t^+)^\perp$.*

Proof (Lemma 12). We prove it by induction on the structure of $t : o$. If $t = F\ t_1 ... t_k$ with $F \in \mathcal{N}$ then $t^\perp = \perp$ and $t^+ = F^{[\![t_1]\!],...,[\![t_k]\!]}\ t_1^{\boldsymbol{\sigma}_1^{\tau_1}} ... t_1^{\boldsymbol{\sigma}_{n_{\tau_k}}^{\tau_k}}$, then $(t^+)^\perp = \perp = t^\perp$.

If $t = a\ t_1 ... t_k$ with $a : o^k \rightarrow o \in \Sigma$ and $t_i : o$ for all i, then $t^+ = a^{[\![t_1]\!],...,[\![t_k]\!]} t_1^+ ... t_k^+$, $t^\perp = a\ t_1^\perp ... t_k^\perp$ and $(t^+)^\perp = a\ (t_1^+)^\perp ... (t_k^+)^\perp$. By induction hypothesis, for all i $(t_i^+)^\perp = t_i^\perp$, then $(t^+)^\perp = a\ t_1^\perp ... t_k^\perp = t^\perp$.

Lemma 13 (Label Conservation in Rewrite Rules). *Given a term $t : \tau = \tau_1 \rightarrow ... \rightarrow \tau_k \rightarrow o \in \mathcal{T}(\Sigma \uplus \mathcal{N})$, such that $t = F\ s_1 ... s_k$ with $F \in \mathcal{N}$, and t is an IO-relevant redex. Note that $t^+ = F^{[\![s_1]\!],...,[\![s_k]\!]}\ s_1^{\boldsymbol{\sigma}_1^{\tau_1}} ... s_k^{\boldsymbol{\sigma}_{n_{\tau_k}}^{\tau_k}}$. If $F\ x_1 ... x_k \rightarrow e \in \mathcal{R}$, let $t' = e_{[\forall i\ x_i \mapsto s_i]}$ and $s = e^+_{\Theta^{\mathcal{V}}\, [x_i^{\boldsymbol{\sigma}_j^{\tau_i}} \mapsto s_i^{\boldsymbol{\sigma}_j^{\tau_i}}]}$ with $\Theta^{\mathcal{V}}(x_i) = [\![s_i]\!]$ for all i (in particular, $t \rightarrow t'$ and $t^+ \rightarrow s$).*

We have $s = (t')^+$.

Proof (Lemma 13). Besides the labeling, by construction, s matches $(t')^+$. Take a subterm e' of e, if one can prove $\Theta^\star \vdash e'_{[\forall i\ x_i \mapsto s_i]} \triangleright \sigma$, then one can prove $\Theta^\star, \Theta^{\mathcal{V}} \vdash e' \triangleright \sigma$, hence s is well labeled, therefore $s = (t')^+$. Then $t^+ \rightarrow^*_{IO} (t')^+$.

Given two terms t, t', we write $t \Rightarrow_{IO} t'$ if t' is obtained by applying in parallel all IO rewrite availables in t. Formally, we define it inductively: if t is an IO-relevant redex and t' is the term obtained by rewriting this redex then $t \Rightarrow_{IO} t'$. If t is not an IO redex and $t = t_1\ t_2$ then $t \Rightarrow_{IO} t'$ if and only if:

- either there exists t'_1, t'_2 such that $t_1 \Rightarrow_{IO} t'_1$ and $t_2 \Rightarrow_{IO} t'_2$, and $t' = t_1\ t_2$,
- or there exists t'_1 such that $t_1 \Rightarrow_{IO} t'_1$ but no t'_2 such that $t_2 \Rightarrow_{IO} t'_2$ and $t' = t'_1\ t_2$,
- or there exists t'_2 such that $t_2 \Rightarrow_{IO} t'_2$ but no t'_1 such that $t_1 \Rightarrow_{IO} t'_1$ and $t' = t_1\ t'_2$.

Notice that if such t' exists then it is unique, and it exists if and only if t contains a redex. The $\cdot \Rightarrow_{IO} \cdot$ relation is known as parallel rewrite, and from a term $t : o$, the unique associated parallel derivation $t \Rightarrow_{IO} t_1 \Rightarrow_{IO} t_2 \Rightarrow_{IO} ...$ leads to the tree $\|G_t\|$.

Lemma 14 (Coincidence of Parallel Derivation). *Given a terms $t \in \mathcal{T}(\Sigma \uplus \mathcal{N})$, and some conjunctive mappings $\sigma_1, ..., \sigma_k$. There exists $t' \in \mathcal{T}(\Sigma \uplus \mathcal{N})$ such that $t \Rightarrow_{IO} t'$, if and only if there exists $s' \in \mathcal{T}(\Sigma' \uplus \mathcal{N}')$ such that $t^{+\sigma_1,...,\sigma_k} \Rightarrow_{IO} s'$. Furthermore, if it is true, then $s' = (t')^{+\sigma_1,...,\sigma_k}$.*

Proof (Lemma 14). The first part of the result comes from the observation that t contains a redex if and only if $t^{+,\sigma_1,...,\sigma_k}$ contains a redex. We prove the second part by induction. If t is an IO-relevant redex, $t^{+\sigma_1,...,\sigma_k}$ is too, and Lemma 13 proves the result. If $t = t_1\ t_2$, t is not an IO-relevant redex a then $t^{+\sigma_1,...,\sigma_k} = t_1^{+\sigma,\sigma_1,...,\sigma_k}\ t_2^{+\boldsymbol{\sigma}_1^\tau} ... t_2^{+\boldsymbol{\sigma}_{n_\tau}^\tau}$ and $t^{+\sigma_1,...,\sigma_k}$ is not an IO-relevant redex. Assume that $t \Rightarrow_{IO} t'$ then,

- either there exists t_1', t_2' such that $t_1 \Rightarrow_{IO} t_1'$ and $t_2 \Rightarrow_{IO} t_2'$, and $t' = t_1'\ 't_2$,
- or there exists t_1' such that $t_1 \Rightarrow_{IO} t_1'$ but no t_2' such that $t_2 \Rightarrow_{IO} t_2'$ and $t' = t_1'\ t_2$,
- or there exists t_2' such that $t_2 \Rightarrow_{IO} t_2'$ but no t_1' such that $t_1 \Rightarrow_{IO} t_1'$ and $t' = t_1\ t_2'$.

By induction hypothesis, $t_i \Rightarrow_{IO} t_i'$ if and only if $t_i^+ \Rightarrow_{IO} t_i'^+$ for $i \in \{1, 2\}$, hence

- either there exists t_1', t_2' such that $t_1^{+\sigma,\sigma_1,...,\sigma_k} \Rightarrow_{IO} t_1'^+$ and $t_2^{+\boldsymbol{\sigma}_j^\tau} \Rightarrow_{IO} t_2'+$ for all j, and $(t')^{+\sigma_1,...,\sigma_k} = (t_1')^{+\sigma,\sigma_1,...,\sigma_k}\ (t_2')^{+\boldsymbol{\sigma}_1^\tau} ... (t_2')^{+\boldsymbol{\sigma}_{n_\tau}^\tau}$,
- or there exists t_1' such that $t_1^{+\sigma,\sigma_1,...,\sigma_k} \Rightarrow_{IO} t_1'^+$ but no s_2' such that $t_2^{+\boldsymbol{\sigma}_j^\tau} \Rightarrow_{IO} s_2'$ for all j, and $(t')^{+\sigma_1,...,\sigma_k} = (t_1')^{+\sigma,\sigma_1,...,\sigma_k}\ (t_2)^{+\boldsymbol{\sigma}_1^\tau} ... (t_2)^{+\boldsymbol{\sigma}_{n_\tau}^\tau}$,
- or there exists t_2' such that $t_2^{+\boldsymbol{\sigma}_j^\tau} \Rightarrow_{IO} (t_2')^{+\boldsymbol{\sigma}_j^\tau}$ but no s_1' such that $t_1^{+\sigma,\sigma_1,...,\sigma_k} \Rightarrow_{IO} s_1'$, and $(t')^{+\sigma_1,...,\sigma_k} = (t_1)^{+\sigma,\sigma_1,...,\sigma_k}\ (t_2')^{+\boldsymbol{\sigma}_1^\tau} ... (t_2')^{+\boldsymbol{\sigma}_{n_\tau}^\tau}$.

Therefore, $t^{+\sigma_1,...,\sigma_k} \Rightarrow (t')^{+\sigma_1,...,\sigma_k}$.

Given a term $t : o$ let $t \Rightarrow_{IO} t_1 \Rightarrow_{IO} t_2 \Rightarrow_{IO} ...$ be the parallel derivation associated to it. Thanks to Lemma 14 we know that the parallel derivation associated to t^+ is $t^+ \Rightarrow_{IO} t_1^+ \Rightarrow_{IO} t_2^+ \Rightarrow_{IO} ...$, then $\|G'_{t^+}\|_{IO}$ is the limit of $(t_i^+)^\perp$ then $(\|G'_{t^+}\|_{IO})^-$ is the limit of $((t_i^+)^\perp)^- = (t_i)^\perp$. Then $\|G'_{t^+}\|_{IO} = \|G_t\|_{IO}$.

Proof ($\|G''\|_{IO} = \|G'\|_{IO}$). Take a term $t \in \mathcal{T}(\Sigma \uplus \mathcal{N})$ we define $void(t) \in \mathcal{T}(\Sigma \uplus \mathcal{N} \uplus \{Void\})$ as the set of termes obtained by substituing some redex r in t such that $\|G_r'\|_{IO} = \perp$ by $Void$. From the definition comes that if $t' \in void(t)$ then $(t')^\perp = t^\perp$.

Given a term $t \in \mathcal{T}(\Sigma \uplus \mathcal{N})$ and an IO derivation associated $t = t_1 \to_{IO} t_2 \to_{IO} ...$ in G' we construct by induction an IO derivation in G'' $t = t_1' \to_{IO} t_2' \to_{IO} ...$ such that for all i $t_i' \in void(t_i)$. The initial step is straightforward:

$t \in void(t)$, Assume that $t'_i \in void(t_i)$, and assume that $t_i = C[F\ t_1...t_k]$ and $t_{i+1} = C[e_{[\forall i\ x_i \mapsto t_i]}]$ with $F\ x_1...x_k \to e \in \mathcal{R}'$. If this redex is a subterm of another one that is transformed by $Void$ in t'_i then we just rewrite this void obtaining $t'_{i+1} = t'_i$ by induction hypothesis, we still have $t'_{i+1} \in void(t_{i+1})$. If this redex is not transformed in t'_i then we rewrite this redex, and either $\|F\ t_1...t_k\|_{IO} = \perp$, in which case the semantics associated contains $q_\perp$ thanks to Theorem **??**, and then $e_{[\forall i\ x_i \mapsto t_i]}$ is still a redex and is transformed to $Void$ in t'_{i+1} or $\|F\ t_1...t_k\|_{IO} \neq \perp$ and no more transformation is added to create t'_{i+1}, in both cases $t'_{i+1} \in void(t_{i+1})$.

This result gives that $\|G'\|_{IO} \sqsubseteq \|G''\|_{IO}$. Since G'' is obtained by from G' changing some redex into other that will produce $\perp$, it is clear that $\|G''\|_{IO} \sqsubseteq \|G'\|_{IO}$, then $\|G'\|_{IO} = \|G''\|_{IO}$.

Proof ($\|G''\| = \|G''\|_{IO}$). We already know that $\|G''\|_{IO} \sqsubseteq \|G''\|$, we just have to show that for all t, if there is $\perp$ at node u in $\|G''_t\|_I O$ then there is $\perp$ at node u in $\|G''_t\|_{IO}$. We show this by induction on the size of u.

If $\|G''_t\|_I O = \perp$. Then $\|G'_t\|_I O = \|G_t\|_I O = \perp$, hence $[\![t]\!] = \perp$, hence the only derivation in G'' from t is $t \to Void \to Void \to ...$, therefore $\|G''\| = \perp$. If $u = ju'$ then let a be the terminal at the root of $\|G''\|_{IO}$, then there exists an IO derivation $t \to^* a\ t_1...t_k$, and $\|G''_{t_j}\|_{IO}$ is equal to the subtree of $\|G''_t\|_{IO}$ rooted at node j. Since $t \to^* a\ t_1...t_k$, $\|G''_{t_j}\|$ is equal to the subtree of $\|G''_t\|$ rooted at node j and by induction hypothesis, since there is $\perp$ at node u' in $\|G''_{t_j}\|_{IO}$, there is $\perp$ at node u' in $\|G''_{t_j}\|$ hence there is $\perp$ at node u in $\|G''_{t_j}\|$.