Available online at [www.sciencedirect.com](http://www.sciencedirect.com/)

[Electronic Notes in Theoretical Computer Science 308 (2014) 245–272](http://dx.doi.org/10.1016/j.entcs.2014.10.014)

[www.elsevier.com/locate/entcs](http://www.elsevier.com/locate/entcs)

Relational Graph Models, Taylor Expansion and Extensionality

Giulio Manzonetto[1](#_bookmark0)*,*[2](#_bookmark1) Domenico Ruoppolo[1](#_bookmark2)*,*[3](#_bookmark3)

*Laboratoire LIPN, CNRS UMR7030 Universit´e Paris 13*

Abstract

We define the class of relational graph models and study the induced order- and equational- theories. Using the Taylor expansion, we show that all λ-terms with the same B¨ohm tree are equated in any relational graph model. If the model is moreover extensional and satisfies a technical condition, then its order-theory coincides with Morris’s observational pre-order. Finally, we introduce an extensional version of the Taylor expansion, then prove that two λ-terms have the same extensional Taylor expansion exactly when they are equivalent in Morris’s sense.

*Keywords:* lambda calculus, linear logic, differential nets, extensional B¨ohm trees, Taylor expansion.

# Introduction

An important problem in the theory of programming languages is to determine when two programs are equivalent. For *λ*-calculus, it has become standard to regard two programs *M* and *N* as equivalent when they are *contextually equivalent* with respect to some fixed set *O* of *observables*. This means that we can plug either *M* or *N* into any context *C*(*−*), i.e. any program with a *hole*, without noticing any difference in the global behaviour: *C*(*M* ) reduces to an observable in *O* exactly when *C*(*N* ) does. Two notable examples are *≡*hnf and Morris’s equivalence *≡*nf [[19](#_bookmark69)] obtained by taking as observables the head normal forms and the *β*-normal forms, respectively. Working with these definitions is difficult because of the quantification over all possible contexts. However, researchers have found alternative characterisations of

these program equivalences based on syntactic trees or denotational models.

For instance, two programs are equivalent with respect to *≡*hnf whenever they have the same Nakajima tree [[20](#_bookmark70)] or, equivalently, when their interpretations coin-

1 This work is partly supported by ANR JCJC Project Coquas 12JS0200601.

2 Email: [giulio.manzonetto@lipn.univ-paris13.fr](mailto:giulio.manzonetto@lipn.univ-paris13.fr)

3 Email: [domenico.ruoppolo@lipn.univ-paris13.fr](mailto:domenico.ruoppolo@lipn.univ-paris13.fr)

<http://dx.doi.org/10.1016/j.entcs.2014.10.014>

1571-0661/© 2014 Elsevier B.V. All rights reserved.

cide in Scott’s model *D∞* [[23](#_bookmark72)]. Similarly, *≡*nf is captured by extensional B¨ohm trees

[[15](#_bookmark65)] and Coppo, Dezani and Zacchi’s filter model *D*cdz [[7](#_bookmark57)].

The idea behind B¨ohm trees, and their extensional versions, is to extract the computational content of a program by representing its output as a possibly infinite tree — the continuity of this representation allows to infer properties of the whole tree by studying its finite approximants. For this reason B¨ohm-like trees and con- tinuous models relied to them via approximation theorems constituted for over forty years the main tools to reason about the behaviour of a program. A limitation of these methods is that they abstract away from the execution process and overlook quantitative aspects such as the time, space, or energy consumed by a computation. The present paper fits in a wider research programme whose aim is to rebuild the traditional theory of program approximations, by replacing it with a mathemat- ical model of resource consumption. The starting point is [[10](#_bookmark60)], where Ehrhard and Regnier propose to analyse the behaviour of a program via its *Taylor expansion*, which is a generally infinite series of “resource approximants”. Such approximants are terms of a *resource calculus* corresponding to a finitary fragment of the differen- tial *λ*-calculus [[8](#_bookmark58)]. Each resource approximant *t* of a *λ*-term *M* captures a particular

choice of the number of times *M* must call its sub-routines during its execution.

Both the differential *λ*-calculus and the Taylor expansion can be naturally in- terpreted in the relational semantics of linear logic [[17](#_bookmark66)]. The first author *et al.* built a relational model *D*ω living in such a semantics [[6](#_bookmark56)] and proved, using standard techniques, that the induced equality is exactly *≡*hnf [[16](#_bookmark67)], just like for Scott’s model *D∞* [[13](#_bookmark63)]. In this paper we provide syntactical and denotational methods based on Taylor expansion that allow to characterise Morris’s equivalence *≡*nf.

First, we introduce the class of *relational graph models* (rgms) of *λ*-calculus, which are the relational analogous of graph models [[3](#_bookmark53)], and describe them as non- idempotent intersection type systems [[21](#_bookmark71)]. This class is general enough to encom- pass all relational models individually introduced in the literature [[6](#_bookmark56),[14](#_bookmark64)], including *D*ω (while Scott’s *D∞* cannot be a graph model since it is extensional). We then show that: (*i*) all rgms satisfy an approximation theorem for resource approximants (Theorem [3.10](#_bookmark76)); (*ii*) in any rgm preserving the polarities of its “empty type” *ω*, *β*- normalisable *λ*-terms can be easily characterized (Lemma [4.3](#_bookmark26)). As a consequence, we get that all extensional rgms preserving *ω*-polarities induce as order-theory Morris’s observational pre-order, and hence *≡*nf as equality (Corollary [4.6](#_bookmark36)). As an instance, we provide the rgm *D*٨ generated by *→* whereis the only atom. It should be compared with the aforementioned filter model *D*cdz, which has the same theory but is more complicated since it has two non-trivially ordered atoms *ϕT ≤ ϕ*٨ and is generated by two equations *ϕT ϕ*٨ *→ ϕT* and *ϕ*٨ *ϕT → ϕ*٨.

Finally, we provide a notion of *extensional Taylor expansion* characterising, like extensional B¨ohm trees, Morris’s equivalence while keeping the quantitative infor- mation. Intuitively, the extensional Taylor expansion of a *λ*-term is the *η*-normal form of its resource approximants. The definition is tricky because the *η*-reduction is meaningless on a *single* resource approximant — one should look at the whole series of approximants to decide whether an element should reduce or not. Our solution is

to define a labeling as a global operation on the series of approximants, and then a local *η*-reduction on labeled terms. Two programs are then *≡*nf-equivalent exactly when they have the same extensional Taylor expansion (Theorem [5.17](#_bookmark50)). We leave for future works a characterisation of Morris’s preorder based on Taylor expansion.

**Basic notations and conventions.** We let **N** denote the set of natural numbers. Given a set *A*, *P*(*A*) (resp. *P*f(*A*)) is the set of all (resp. finite) subsets of *A* and *M*f(*A*) is the set of all finite multisets over *A*. Finite multisets are represented as unordered lists *m* = [*α*1*,..., α*n] with repetitions, [] being the empty multiset.

Given a reduction *→*r we write →r (=r) for its transitive and reflexive (and symmetric) closure. A term *t* has an r-normal form nfr(*t*), if *t* →r nfr(*t*) */→*r.

**N.B.** Unless otherwise stated, throughout the paper we suppose that all operators

*F* : *A → B* are extended to *P*(*A*) in the natural way: *F* (*a*)= *{F* (*α*) *| α ∈ a}*.

# Lambda Calculus and B¨ohm Trees

We will generally use the notation of Barendregt’s classic work [[2](#_bookmark51)] for *λ*-calculus. Let us fix an infinite set Var of variables. The set Λ of *λ-terms* is defined by:

Λ: *M, N, P* ::= *x | λx.M | MN* for all *x ∈* Var*.*

The set fv(*M* ) of *free variables* of *M* and the *α*-conversion are defined as usual, see [[2](#_bookmark51), Ch. 1*§*2]. A *λ*-term *M* is *closed* if fv(*M* )= *∅*. We denote by Λo the set of closed *λ*-terms. From now on, *λ*-terms will be considered up to *α*-conversion.

Given two *λ*-terms *M, N* we denote by *M{N/x}* the capture-free substitution of *N* for all free occurrences of *x* in *M* . The *β*- and *η*-reductions are given for granted. Concerning specific *λ*-terms, we fix the identity **I** = *λx.x*, its *η*-expansion **1** = *λxy.xy*, the paradigmatic looping term Ω = ΔΔ where Δ = *λx.xx*, Turing’s fixpoint combinator Θ = *λf.*Θf Θf where Θf = *λx.f* (*xx*) and **J** = Θ(*λzxy.x*(*zy*)) a term

reducing to an infinite *η*-expansion of **I**.

A *λ*-term *M* is called *solvable* if it has a *head normal form* (*hnf*, for short), that is if *M* →β *λx*1 *... x*n*.yN*1 *··· N*k (for *n, k ≥* 0); otherwise *M* is called *unsolvable*.

Given a *context C*(*−*), i.e. a *λ*-term with a *hole* denoted by (*−*), we write *C*(*M* ) for the *λ*-term obtained from *C* by substituting *M* for the hole possibly with capture of free variables in *M* . Given *O⊆* Λ, the *O-observational pre-order* is defined by:

*M ±& N ⇐⇒ ∀C*(*−*) *. C*(*M* ) →β *Mj ∈O* entails *C*(*N* ) →β *Nj ∈ O.*

The induced *equivalence M ≡& N* is defined as *M ±& N* and *N ±& M* . To obtain

*Morris’s pre-order ±*nf and *equivalence ≡*nf just take as *O* the set of *β*-nfs [[19](#_bookmark69)].

The *B¨ohm tree* BT(*M* ) ofa *λ*-term *M* is defined coinductively: if *M* is unsolvable then BT(*M* )= *⊥*; if *M* is solvable, then *M* →β *λx*1 *... x*n*.yN*1 *··· N*k and

BT(*M* )= *λx*1 *... x*n*.y*

BT(*N*1) *···* BT(*N*k)

Such a definition is sound in the sense that *M* =β *N* entails BT(*M* ) = BT(*N* )*.* Examples of B¨ohm trees are: BT(**I**)= **I**, BT(**1**)= **1**, BT(Δ) = Δ, BT(Ω) = *⊥*,

BT(*λx.y*Ω) = *λx.y* BT(**J**)= *λxz*0*.x* BT(Θ) = *λf.f*

*⊥ λz*1*.z*0 *f*

*λz*2*.z*1 *f*

Given two B¨ohm trees *T, T j* we set *T ≤⊥ T j* if and only if *T* results from *T j* by replacing some subtrees with *⊥*. The set *N* of *ﬁnite approximants* is the set of *λ*-terms possibly containing *⊥* inductively defined as follows: *⊥∈ N* ; if *a*i *∈N* for *i* = 1*,...,n* then *λ→x.ya*1 *··· a*n *∈ N* . Hereafter we will confuse finite B¨ohm trees with normal approximants. Notice that the set of all finite approximants of a B¨ohm tree *T* , given by *T∗* = *{a ∈N | a ≤**⊥ T}*, is an ideal with respect to *≤⊥* [[1](#_bookmark52), *§*2.3].

A *λ-theory* is any congruence on Λ containing =β. A *λ*-theory is: *extensional* if it contains =η; *sensible* if it equates all unsolvables. We denote by: *λβη* the least extensional *λ*-theory; *B* the *λ*-theory equating all *λ*-terms having the same B¨ohm tree; *Bη* the least *λ*-theory containing *B* and *λβη* ; *H*+ (resp. *H∗*) the *λ*-theory characterizing *≡*nf (resp. *≡*hnf). From [[2](#_bookmark51), Thm. 17.4.16] we get *B* Ç *Bη* Ç *H*+ Ç *H∗*.

# Resource Calculus and Taylor Expansion

We briefly recall Ehrhard’s *resource calculus* [[9](#_bookmark59)], using the syntax proposed by Tran- quilli in [[24](#_bookmark73)]. We are considering here the promotion-free fragment of [[24](#_bookmark73)].

**Syntax.** The set Λr of *resource terms* and the set Λb of *bags* are defined by: Λr : *s, t* ::= *x | λx.t | tb* Λb : *b* ::= [*s*1*,..., s*n] where *n ≥* 0*.* (1)

Resource terms are in functional position, while bags are in argument position and represent unordered lists of resource terms. Intuitively, in a term of shape *t*[*s*1*,..., s*n] each *s*i is a linear resource, that is *t* cannot duplicate nor erase it.

We will deal with bags as if they were multisets presented in multiplicative notation: 1 is the empty bag and *b*1 *· b*2 is the multiset union of *b*1 and *b*2.

We use the power notation [*s*k] for the bag [*s,..., s*] containing *k* copies of *s*.

The *α*-equivalence and the set fv(*t*) of free variables of *t* are defined as for the ordinary *λ*-calculus. Resource terms and bags are considered up to *α*-equivalence.

As a syntactic sugar, we extend all the constructors of the grammar ([1](#_bookmark5)) as pointwise operations on (possibly infinite) sets of resource terms or bags. That is, given O *⊆* Λr and B*,* B*j ⊆* Λb we use the following notations: *λx.*O = *{λx.t | t ∈* O*}*, OB = *{tb | t ∈* O*,b ∈* B*}*, [O]= *{*[*t*] *| t ∈* O*}* and B *·* B*j* = *{b · bj | b ∈* B*, bj ∈* B*j}*.

Observe that, in the particular case of empty set, we get *λx.∅* = *∅*, *t∅* = *∅*,

*∅b* = *∅*, [*∅*]= *∅* and *∅· b* = *∅*. Hence, *∅* annihilates any resource term or bag.

This kind of meta-syntactic notation is discussed thoroughly in [[10](#_bookmark60)].

**Reductions.** Given a relation *→*r*⊆* Λr *× P*f(Λr) its *context closure* is the least relation in *P*f(Λr) *× P*f(Λr) such that, when *t →*r O, we have:

*λx.t →*r *λx.*O*, tb →*r O*b, s*([*t*] *· b*) *→*r *s*([O] *· b*)*, {t}∪* S *→*r O *∪* S*.*

We say that *t ∈* Λr *is in* r*-normal form* if there is no O such that *t →*r O. When

*→*r is confluent, nfr(*t*) *∈ P*f(Λr) denotes the unique r-normal form of *t*, if it exists. The *degree of x in t*, written degx(*t*), is the number of free occurrences of *x* in *t*.

A *β-redex* is a resource term of the shape (*λx.t*)[*s*1*,..., s*k] and its *contractum* is a

finite set of resource terms: when degx(*t*)= *k*, it is the set of all possible resource terms obtained by linearly replacing each free occurrence of *x* in *t* by exactly one of the *s*i’s; otherwise, when degx(*t*) */*= *k*, it is just *∅*.

Formally, we define *→*β as the context closure of:

(*λx.t*)[*s ,...,s* ] *→*

⎧⎨ Sp*∈*S*k t{s*p(1)*/x*1*,..., s*p(k)*/x*k*}* if degx(*t*)= *k,*

1 k β

⎩ *∅* otherwise.

where Sk is the group of permutations of *{*1*,..., k}* and *x*1*,..., x*n is an arbitrary enumeration of the free occurrences of *x* in *t*. Note that *β*-reduction is strongly normalizing (SN, for short) on *P*f(Λr), since whenever *t →*β O the size of *t* is strictly bigger than the size of each resource term in O. Moreover, *β*-reduction is weakly confluent, and therefore confluent by Newman’s lemma.

**Theorem 2.1** *The β-reduction is strongly normalizing and confluent on P*f(Λr)*.*

In the resource calculus there is no sensible notion of *η*-reduction on *P*f(Λr).

**Taylor expansion**. The *Taylor expansion* of a *λ*-term, as defined in [[8](#_bookmark58),[10](#_bookmark60)], is a translation developing every *λ*-calculus application as an infinite series of resource applications with rational coefficients. For our purpose it is enough to consider a simplified version *T* (*−*):Λ *→ P*(Λr) corresponding to the support [4](#_bookmark7) of the actual Taylor expansion; that is, we consider possibly infinite sets of resource *λ*-terms.

**Definition 2.2** The *Taylor expansion T* (*M* ) *⊆* Λr of a *λ*-term *M* is defined by:

*T* (*x*)= *x, T* (*λx.M* )= *λx.T* (*M* )*, T* (*MN* )= *T* (*M* )*M*f(*T* (*N* ))*.*

The Taylor expansion is extended to finite approximants in *N* by setting *T* (*⊥*)= *∅*, and to B¨ohm trees *T* by setting *T* (*T* )= *{T* (*a*) *| a ∈ T∗}*.

S

Some examples of Taylor expansions of ordinary *λ*-terms are:

*T* (**I**)= *{***I***}, T* (Δ) = *{λx.x*[*x*n] *| n ≥* 0*}, T* (*λy.xyy*)= *{λy.x*[*y*n][*y*k] *| n, k ≥* 0*},*

4 I.e., the set of those resource terms appearing in the series with a non-zero coefficient.

*T* (Ω) = *{*(*λx.x*[*x*n0 ])[*λx.x*[*x*n1 ]*,..., λx.x*[*x*n*k* ]] *| k, n*0*,..., n*k *≥* 0*},*

*T* (Θ) = *{λf.*(*λx.f* [*x*[*x*n1 ]*,..., x*[*x*n*k* ]])[*λx.f* [*x*[*x*n1*,*1 ]*,..., x*[*x*n1*,k*1 ]]*,...,*

*λx.f* [*x*[*x*n*h,*1 ]*,..., x*[*x*n*h,kh* ]]] *| k, n*i*, h, n*i,j *≥* 0*},*

*T* (**J**) = *{t*[*λzxy.x*[*z*[*y*n1*,*1 ]*,..., z*[*y*n1*,k*1 ]]*,...,*

*λzxy.x*[*z*[*y*n*h,*1 ]*,..., z*[*y*n*h,kh* ]]] *| t ∈T* (Θ)*, h, k*i*, n*i,j *≥* 0*}.*

From the examples above it is clear that if a *λ*-term *M* has a *β*-redex, then there are resource terms *t ∈T* (*M* ) having *β*-redexes too. However, by Theorem [2.1](#_bookmark6), each *t* has a unique *β*-nf and we can always compute nfβ(*T* (*M* )) = *{*nfβ(*t*) *| t ∈T* (*M* )*}*. For instance: *T* (**I**), *T* (Δ) and *T* (*λy.xyy*) are already *β*-normal, while nfβ(*T* (Ω)) = *∅*.

S

**Lemma 2.3** *Let a ∈N and M ∈* Λ*, then T* (*a*) *⊆T* (BT(*M* )) *entails a ∈* BT(*M* )*∗.* The following results proved in [[9](#_bookmark59)] show the strong relationship between the

B¨ohm tree of a *λ*-term, and its Taylor expansion.

**Theorem 2.4** *For every λ-term M,* nfβ(*T* (*M* )) = *T* (BT(*M* ))*.*

**Corollary 2.5** *For all M, N ∈* Λ*,* BT(*M* )= BT(*N* ) *iff* nfβ(*T* (*M* )) = nfβ(*T* (*N* ))*.* Using Theorem [2.4](#_bookmark9), we can easily calculate further examples:

nfβ(*T* (Θ)) = *{λf.f* 1*, λf.f* [(*f* 1)n]*, λf.f* [*f* [(*f* 1)n1 ]*,...,f* [(*f* 1)n*k* ]]*,... },*

nfβ(*T* (**J**)) = *{λxz*0*.x*1*, λxz*0*.x*[(*λz*1*.z*01)n]*,... }.*

# Relational Graph Models and Intersection Types

In this section we introduce the class of *relational graph models* (rgm, for short); some examples of such models were individually studied in [[14](#_bookmark64)].

* 1. *Relational Graph Models*

We call rgms *relational* because they are (linear) reflexive objects in the ccc **MRel** [[6](#_bookmark56)], the Kleisli category of **Rel** with respect to the comonad *M*f(*−*). In **MRel** the objects are all the sets, a morphism *f ∈* **MRel**(*A, B*) is any relation between *M*f(*A*) and *B*, and the exponential object *A ⇒ B* is given by *M*f(*A*)*×B*. Any func- tion *f* : *A → B* can be sent to *f† ∈* **MRel**(*A, B*) by setting *f†* = *{*([*a*]*,f* (*a*)) *| a ∈ A}*.

**Definition 3.1** A *relational graph model D* = (*D, i*) is given by an infinite set *D*

and a total injection *i* : *M*f(*D*) *× D → D*. *D* is *extensional* when *i* is bijective.

Every rgm *D* = (*D, i*) induces a reflexive object (*D, i†,* (*i−*1)*†*), i.e. *D ⇒ D* D *D* since *i†*; (*i−*1)*†* = IdD*⇒*D. When *D* is moreover extensional we also have (*i−*1)*†* ; *i†* = idD. These reflexive objects are all *linear* in the sense of [[17](#_bookmark66)] and live in a differential ccc, they are therefore sound models of the resource calculus as well (Theorem [3.8](#_bookmark19)). Rgms, just like the regular ones [[3](#_bookmark53)], can be built by performing the free comple- tion of a partial pair. A *partial pair A* is a pair (*A, j*) where *A* is a non-empty set of elements (called *atoms*) and *j* : *M*f(*A*) *× A → A* is a partial injection. We say

that *A* is *extensional* when *j* is a bijection between dom(*j*) and *A*. Wlog., we will only consider partial pairs *A* whose underlying set *A* does not contain any pair.

S

**Definition 3.2** The *completion A* of a partial pair *A* is the pair (*A, j*) defined as: *A* = n*∈***N** *A*n, where *A*0 = *A* and *A*n+1 = ((*M*f(*A*n) *× A*n) *−* dom(*j*)) *∪ A* ; the function *j* is given by *j*(*a, α*)= *j*(*a, α*) if (*a, α*) *∈* dom(*j*), *j*(*a, α*)= (*a, α*) otherwise.

Note that, for every rgm *D* we have *D* = *D* (up to isomorphism).

**Proposition 3.3** *If A is a partial pair, then A is an rgm. When A is extensional, also A is extensional.*

**Proof** The proof of the fact that *A* is an rgm is analogous to the one for regular graph models [[3](#_bookmark53)]. It is easy to check that when *j* is bijective, also *j* is. *2*

**Example 3.4** We define the relational analogues of:

* Engeler’s model [[11](#_bookmark61)]: *E* = (**N***, ∅*), first defined in [[14](#_bookmark64)],
* Scott’s model [[23](#_bookmark72)]: *D*ω = (*{ε}, {*([]*, ε*) *≡→ ε}*), first defined (up to iso) in [[6](#_bookmark56)],
* Coppo, Dezani and Zacchi’s model [[7](#_bookmark57)]: *D*٨ = (*{ }, {*([]*,* ) *≡→ }*). Notice that *D*ω and *D*٨ are extensional, while *E* is not.
  1. *Non-Idempotent Intersection Type Systems*

As discussed thoroughly in [[21](#_bookmark71)], the choice of presenting a relational model as a reflexive object or as a non-idempotent intersection type system is more a matter of taste rather than a technical decision. Here we provide the latter presentation.

Let *A* be a partial pair and *D* be its completion. The set T*D* of *types* and the set I*D* of *non-idempotent intersections* are defined by mutual induction (for *α ∈ A*):

T*D* : *σ, τ* ::= *α | μ → σ* I*D* : *μ, ν* ::= *ω | σ | σ ∧ μ*

Note that types are (unary) intersections while the converse does not hold; indeed intersections may only appear at the left-hand side of an arrow. Thus *ω* is not a type, it denotes the empty intersection and is therefore its neutral element (*μ ∧ ω* = *μ*).

Accordingly, we write *∧*n *σ*n for *σ*1 *∧· · ·∧ σ*n when *n ≥* 1, and for *ω* when *n* = 0.

i=1

Types will be considered up to associativity and commutativity of *∧* and neutrality of *ω*, while we assume that the intersection is *not* idempotent, that is *σ ∧ σ /*= *σ*.

Every *σ ∈* T*D* (*μ ∈* I*D*) corresponds to an element *σ•* of *D* (*μ•* of *M*f(*D*)) defined as *α•* = *α*, (*μ → τ* )*•* = *i*(*μ•,τ•*) and (*σ*1 *∧· · ·∧σ*n)*•* = [*σ•, ..., σ•* ]. Hence, the model

1 n

*D* induces a congruence on the intersection types: *σ D τ* if and only if *σ•* = *τ•*.

An *environment* is a map Γ : Var *→* I*D* such that dom(Γ) = *{x |* Γ(*x*) */*= *ω}* is finite. We write *x*1 : *μ*1*,..., x*n : *μ*n for the environment Γ such that Γ(*x*i)= *μ*i and Γ(*y*)= *ω* for all *y ∈/ →x*. The environment mapping all variables to *ω* is denoted by

*∅*, or just omitted as in Example [3.6](#_bookmark14). The intersection Γ1 *∧* Γ2 and the equivalence Γ1 *D* Γ2 of two environments are defined pointwise; note that Γ *∧∅* = Γ.

**Definition 3.5** The interpretation of *M ∈* Λ (or *M ∈ N* ) in *D* is defined as:

*x* : *σ ▶D x* : *σ*

var

Γ*,x* : *μ ▶D M* : *σ*

Γ *▶D λx.M* : *μ → σ* lam

Γ *▶D M* : *τ σ D τ*

Γ *▶D M* : *σ*

eq

Γ0 *▶D M* : *∧*n *σ*i *→ τ* Γi *▶D N* : *σ*i for *i* = 1*,...,n*

i=1 app

Γ0 *∧* (*∧*n Γi) *▶D MN* : *τ*

i=1

(a) Non-idempotent intersection type system for Λ and *N* .

Γ0 *▶D t* : *∧*n *σ*i *→ τ* Γi *▶D s*i : *σ*i for *i* = 1*,...,n*

i=1 app*j*

i=1

(b) Non-idempotent intersection type system for Λ*r*.

Γ0 *∧* (*∧*n Γi) *▶D t*[*s*1*,..., s*n]: *τ*

**Figure 1:** The intersection type systems for Λ, *N* and Λr . The other rules for typing Λr are analogous to (var), (lam), (eq) of Figure [1(a)](#_bookmark12) and are omitted.

J*M* )*Q* = *{*(Γ*, σ*) *|* Γ *▶Q M* : *σ},* where the type system *▶Q* is given in Fig. [1(a)](#_bookmark12)*.*

The definition of J*t*)*Q* for *t ∈* Λr is analogous, using the rules of Fig. [1(b)](#_bookmark13). Note that

*▶Q* also works for terms in *N* : *⊥* is not typable, but e.g. *▶Q λx.x⊥* : (*ω → τ* ) *→ τ* .

**Example 3.6** Let *D* be any rgm. Then we have: J**I**)*Q* = *{σ | σ τ → τ, τ ∈* T*Q}*, J**1**)*Q* = *{σ | σ* (*μ → τ* ) *→ μ → τ, τ ∈* T*Q,μ ∈* I*Q}*, J**J**)*Q* = *{σ | σ* (*ω → τ* ) *→* *ω → τ, τ ∈* T*Q}*, J*λx.x*Ω)*Q* = *{σ | σ* (*ω → τ* ) *→ τ, τ ∈* T*Q}*, JΩ)*Q* = *∅*. It follows that J**I**) = J**1**) in both *D*ω and *D*٨, but J**I**)*Qu* = J**J**)*Qu* , while *∈* J**I**)*Qs −* J**J**)*Qs* .

When *D* is clear from the context we simply write, *▶* and J*−*). Note that Γ *▶ M* : *σ* implies dom(Γ) *⊆* fv(*M* ) and Γ*j ▶ M* : *σj* for ΓΓ*j* and *σ σj* [[21](#_bookmark71)].

**Theorem 3.7 (Inversion Lemma, cf. [**[**21**](#_bookmark71)**])** *Let D be an rgm.*

1. Γ *▶ x* : *σ entails* Γ= *x* : *τ for τ σ,*
2. Γ *▶ λx.M* : *σ if and only if* Γ*,x* : *μ ▶ M* : *τ for some μ → τ σ,*
3. Γ *▶ MN* : *σ entails that* Γ= Γ0 *∧*(*∧*n

i=1

Γi) *for some n ≥* 0*,* Γ0 *▶ M* : *∧*n

*σ*i *→*

*σ and* Γi *▶ N* : *σ*i*.*

i=1

*For resource λ-terms an analogous statement holds, where* (*iii*) *is replaced with:*

*(iii’)* Γ *▶ t*[*s*1*,..., s*n]: *σ entails* Γ= Γ0*∧*(*∧*n

i=1

Γi)*,* Γ0 *▶ t* : *∧*n

*σ*i *→ σ and* Γi *▶ s*i : *σ*i*.*

**Theorem 3.8** *Let D be an rgm, then for* Λ *and* Λr*:*

i=1

1. *Substitution lemma, subject reduction and subject expansion hold in ▶Q.*
2. *The interpretation* J*−*)*Q is* sound *with respect to* =β*.*

**Proof** (*i*) is proved in [[21](#_bookmark71)] for Λ and in [[17](#_bookmark66)] for relational models of Λr.

(*ii*) follows from (*i*). *2*

The *λ-theory* and the *order theory* induced by *D* are given by Th(*D*) =

*{*(*M, N* ) *|* J*M* ) = J*N* )*}* and Th*≤*(*D*) = *{*(*M, N* ) *|* J*M* ) *⊆* J*N* )*}*, respectively. We write *D |*= *M* = *N* if (*M, N* ) *∈* Th(*D*), and *D |*= *M ≤ N* if (*M, N* ) *∈* Th*≤*(*D*).

A model *D* is *O-inequationally fully abstract* when *D |*= *M ≤ N* if and only if

*M ±& N* , and *O-fully abstract* when *D |*= *M* = *N* if and only if *M ≡& N* .

**Lemma 3.9** *If D is an extensional rgm, then λβη ⊆* Th(*D*)*.*

**Proof** The equivalence between Γ *▶ M* : *σ* and Γ *▶ λx.Mx* : *σ* when *x ∈/* fv(*M* ) follows by induction on *σ* using the fact that *α μ → τ* for every atomic type *α*.*2*

As a consequence, the *λ*-theories induced by rgms and by regular graph models are different, since no graph model is extensional. For instance, the *λ*-theory of *D*ω, the relational analogue of Scott’s *D∞*, is *H*٨ [[16](#_bookmark67)]. That is *D*ω is hnf-fully abstract. While approximation theorems for B¨ohm trees and idempotent intersection type systems are usually proved through reducibility techniques, the following one for Taylor expansion and rgms can be proved by induction on the type derivation using

the subject reduction (Theorem [3.8](#_bookmark19)) and the SN of Λr (Theorem [2.1](#_bookmark6)).

**Theorem 3.10 (Approximation Theorem)** *Let M be a λ-term. Then*

Γ *▶ M* : *σ if and only if there exists t ∈T* (*M* ) *such that* Γ *▶ t* : *σ.*

*Therefore* J*M* ) = J*T* (*M* ))*.*

**Corollary 3.11** *For all rgms D we have that B ⊆* Th(*D*)*. In particular* Th(*D*) *is* *sensible and* J*M* )*Q* = *∅ for all unsolvable λ-terms M.*

St*∈f* (M)

**Proof** From Theorem [3.10](#_bookmark76) we have J*M* ) = J*T* (*M* )) = J*t*). By subject reduction for Λr (Theorem [3.8](#_bookmark19)) this is equal to Jnf (*t*)), which is equal to t*∈f* (BT(M))J*t*) = J*T* (BT(*M* ))), by Theorem [2.4](#_bookmark9). Therefore, whenever BT(*M* ) = BT(*N* ) we get J*M* ) = J*T* (BT(*M* ))) = J*T* (BT(*N* ))) = J*N* ). *2*

[S](#_bookmark9)t*∈f* (M) β

S

# Full Abstraction for Morris’s Observational Preorder

This section is devoted to show that every extensional rgm *D* satisfying the condition of Definition [4.1](#_bookmark24) — in particular *D*٨ — is (inequationally) fully abstract with respect to Morris’s pre-order *±*nf. Rather than working directly with *±*nf, and building separating contexts, we use Levy’s notion of *extensional B¨ohm tree*

BTe(*M* )= *{*nfη(*a*) *| a ∈* BT(*Mj*)*∗, Mj* →η *M}.*

Indeed, it is well known that *M ±*nf *N* exactly when BTe(*M* ) *⊆* BTe(*N* ) [[12](#_bookmark62)] and that two *λ*-terms have the same extensional B¨ohm tree when their B¨ohm trees are equal up to (possibly infinitely many) *η*-expansions of *ﬁnite depth*. These trees are therefore different from Nakajima trees: for instance **I** *∈* BTe(**I**) *−* BTe(**J**).

Examples of extensional B¨ohm trees are: BTe(**1**)= BTe(**I**),

BTe(**I**)= *{⊥,* **I***, λxz*0*.x⊥, λxz*0*.x*(*λz*1*.z*0(*λz*2*.z*1*⊥*))*,... },* BTe(**J**)= BTe(**I**) *− {***I***},*

BTe(*λy.xyy*)= *{⊥, x⊥, λy.xyy, λy.xy⊥,.* *},* BTe(*x*Ω) = BTe(*λy.xyy*)*−{λy.xyy}.*

Given a polarity *p ∈ {*+*, −}*, we define inductively for all types *σ* the relations

*ω ∈*p *σ* and *ω ∈ч*p *σ*, where *¬p* is the opposite polarity, as: (*i*) *ω ∈— μ → τ* if

*μ* = *ω*; (*ii*) if *ω ∈*p *τ* then *ω ∈*p *μ → τ* ; (*iii*) if *ω ∈ч*p *τ* then *ω ∈*p *τ ∧ μ → τj*. When *ω ∈*+ *σ* (*ω ∈— σ*) we say that *ω occurs positively* (*negatively* ) in *σ*. We write *ω ∈/*+ *σ* (*ω ∈/— σ*) if *ω* does not occur positively (negatively) in *σ*. These notions extend to intersections in the obvious way, for instance *ω ∈*p *σ*1 *∧ ··· ∧ σ*n if *ω ∈*p *σ*i for some *i*.

**Definition 4.1** An rgm *D preserves ω-polarities* whenever *ω ∈*p *σ* and *σ τ* entail

*ω ∈*p *τ* , for all *σ, τ ∈* T*Q* and *p ∈ {*+*, −}*.

For instance *E* and *D*٨ preserve *ω*-polarities, while *D*ω does not because *ω ∈*+ (*ω → ε*) *→ ε ε → ε* but *ω ∈/*+ *ε → ε*. Note that, if an rgm *D* preserve *ω*-polarities, then we also have that *ω ∈/*p *σ* and *σ τ* entail *ω ∈/*p *τ* (where *p ∈ {*+*, −}*).

**Proposition 4.2** *Let A be a partial pair such that, for all m ∈ M*f(*A*) *and α ∈ A,*

(*m, α*) *∈* dom(*j*) *entails that m /*= []*. Then A preserves ω-polarities.*

**Lemma 4.3** *Let M ∈* Λ*. The following are equivalent:*

1. *M has a normal form,*
2. *there is a ∈* BT(*M* )*∗ that does not contain ⊥,*
3. *there is t ∈* nfβ(*T* (*M* )) *that does not contain the empty bag* 1*,*
4. *in every rgm D preserving ω-polarities,* Γ *▶Q M* : *σ for some environment* Γ

*and type σ such that ω ∈/*+ *σ and ω ∈/—* Γ *(that is ω ∈/—* Γ(*x*) *for all x ∈* Var*).*

**Proof** [Sketch] ([*i*](#_bookmark27) *⇐⇒* [*ii*](#_bookmark28)) is trivial and ([*ii*](#_bookmark28) *⇐⇒* [*iii*](#_bookmark29)) follows from Theorem [2.4](#_bookmark9). ([*iii*](#_bookmark29) *⇒* [*iv*](#_bookmark30)) One proves by induction on the *β*-normal *t* that Γ *▶ t* : *σ* holds for

some Γ*,σ* such that *ω ∈/—* Γ and *ω ∈/*+ *σ*. Then one concludes by subject expansion for Λr and the approximation theorem (Theorem [3.10](#_bookmark76)).

([*iv*](#_bookmark30) *⇒* [*iii*](#_bookmark29)) By the approximation theorem and subject reduction for Λr there is *t ∈* nfβ*T* (*M* ) such that Γ *▶ t* : *σ* is derivable for some Γ*,σ* satisfying *ω ∈/—* Γ and *ω ∈/*+ *σ*. Then, using Theorem [3.7](#_bookmark15) and the preservation of *ω*-polarities, one proves by induction on the structure of normal form of *t* that it does not contain 1. *2*

Notice that in the model *D*ω, which does not preserve *ω*-polarities, the above lemma does not hold. For instance, *ω ∈/*+ *ε → ε ∈* J**J**)*Qu* , but **J** is not normalizing.

In Coppo, Dezani and Zacchi’s model *D*cdz presented in [[7](#_bookmark57)], there is an atomic type *ϕ*٨ (resp. *ϕT*) characterizing the terms having a *β*-nf (resp. persistent *β*-nf).

In the model *D*٨ the typecaptures those *λ*-terms *M ∈* Λo having a normal form that is “linear”. A *λ*-term *M* is called *linear* whenever: (*i*) every *y ∈* fv(*M* ) occurs once in *M* ; (*ii*) every subterm *λx.N* of *M* is such that *x* occurs once in *N* .

**Lemma 4.4** *Let M ∈* Λ *and* Γ= *x*1 : *,..., x*n : *. Then* Γ *▶Qs M* : *if and only if M has a linear β-normal form and* fv(nfβ(*M* )) = dom(Γ)*.*

We now prove the main results of the section.

**Theorem 4.5** *Let D be an extensional rgm preserving ω-polarities. The following are equivalent (for M, N ∈* Λo*):*

1. *D |*= *M ≤ N,*
2. *M ±*nf *N,*
3. BTe(*M* ) *⊆* BTe(*N* )*.*

**Proof** ([*i*](#_bookmark33) *⇒* [*ii*](#_bookmark34)) Suppose J*M* ) *⊆* J*N* ) and consider a context *C*(*−*) such that *C*(*M* ) has a normal form. By Lemma [4.3](#_bookmark26) there is *σ ∈* J*C*(*M* )) such that *ω ∈/*+ *σ*. Since J*−*) is contextual we have J*C*(*M* )) *⊆* J*C*(*N* )), therefore *σ ∈* J*C*(*N* )) and, by applying Lemma [4.3](#_bookmark26) again, we conclude that *C*(*N* ) has a normal form.

([*ii*](#_bookmark34) *⇐⇒* [*iii*](#_bookmark35)) See Hyland’s original paper [[12](#_bookmark62)], or [[22](#_bookmark74)] for a cleaner proof. ([*iii*](#_bookmark35) *⇒* [*i*](#_bookmark33)) We have: J*M* ) = *∪*M*′*→ M J*Mj*) by Lemma [3.9](#_bookmark22)

*η*

= *∪*M*′*→ M J*T* (*Mj*)) by Theorem [3.10](#_bookmark76)

*η*

= *∪*M*′*→ M Jnfβ*T* (*Mj*)) by Theorem [3.8](#_bookmark19)(*ii*) for Λr

*η*

= *∪*M*′*→ M JBT(*Mj*)*∗*) by Theorem [2.4](#_bookmark9)

*η*

= *∪*M*′*→ M JnfηBT(*Mj*)*∗*) by Lemma [3.9](#_bookmark22)

*η*

= JBTe(*M* )) by definition of BTe(*M* )*.*

Thus BTe(*M* ) *⊆* BTe(*N* ) entails J*M* ) = JBTe(*M* )) *⊆* JBTe(*N* )) = J*N* ). *2*

**Corollary 4.6 (Full abstraction)** *Every extensional rgm D respecting ω-polari- ties has order-theory* Th*≤*(*D*)= *{*(*M, N* ) *| M ±*nf *N} and λ-theory* Th(*D*)= *H*+*.*

# Extensional Taylor Expansion and *η*-Trees

We introduce the notion of *extensional Taylor expansion T* η(*M* ) *of a λ-term M* and prove that it is equal to the Taylor expansion of the extensional B¨ohm tree of *M* (Theorem [5.15](#_bookmark48)). This result is the analogue of Theorem [2.4](#_bookmark9). As a byproduct, we obtain a new syntactical characterization of *≡*nf (Corollary [5.17](#_bookmark50)).

For technical reasons, we work with an alternative notion of extensional B¨ohm tree of *M* , that will be denoted by BTη(*M* ). Rather than producing a set of *η*- normal approximants, BTη(*−*) gives an actual (possibly infinite) *η*-normal tree.

The *η-normal form η*(*T* ) *of a B¨ohm tree T* is defined coinductively: *η*(*⊥*)= *⊥* and

⎧⎪ *λx*1 *... x*n*—*1*.y*

*η*

*λx*1

⎪⎨

m*—*1

*... x*n*.y*

*T*

1

*··· T*

If *x*n *∈/* fv(*yT*1 *··· T*m*—*1)

*T*1 *··· T*m

*η*

=

*λx*1 *... x*n*.y*

⎪⎪⎩ 

*T*m *∈ N* , i.e. it is finite

and *T*m →η *x*n,

*η*(*T*1) *··· η*(*T*m)

otherwise.

Therefore, we define the *B¨ohm η-tree* BTη(*M* ) *of a λ-term M* as *η*(BT(*M* )).

Examples of B¨ohm *η*-trees are: BTη(**J**) = BT(**J**), BTη(*λy.xyy*) = *λy.xyy*, BTη(*λxy*1*y*2*.x*(*λz*1*.y*1(*λz*2*.z*1(*λz*3*.z*2*z*3))*y*2)= BTη(**I**)= **I***,* and BTη(*λy.x⊥y*)= *x⊥.*

The notions of BTη(*—*) and BTe(*—*) are equivalent in the sense that, for all *M, N ∈* Λ, BTe(*M* ) = BTe(*N* ) if and only if BTη(*M* ) = BTη(*N* ) [[25](#_bookmark75),[15](#_bookmark65)]. On the other hand, BTe(*M* ) *⊆* BTe(*N* ) is not equivalent to BTη(*M* ) *≤⊥* BTη(*N* ).

E.g. BTe(*x⊥*) *⊆* BTe(*λy.xyy*) but BTη(*x⊥*)= *x⊥ /≤⊥ λy.xyy* = BTη(*λy.xyy*).

* 1. *Extensional Taylor Expansion*

In order to obtain the analogue of Ehrhard and Regnier’s Theorem [2.4](#_bookmark9) in the exten- sional setting, the extensional Taylor expansion of *M* should be the *η*-normal form of nfβ*f* (*M* ), just like BTη(*M* ) is the *η*-normal form of BT(*M* ).

The problem is that defining an *η*-reduction on *P*(nfβ(Λr)) is no easy task. Consider for instance the *na¨ıve* definition *→*η= *∪*k*≥*0(*→*ηk) where *λx.t*[*x*k] *→*ηk *t* if *x ∈/* fv(*t*). This correctly reduces *f* (*λy.xy*) = *{λy.x*[*y*k] *| k ≥* 0*}* to *{x}*, but the fact that *λy.x*1 *→*η0 *x* is a problem, since *λy.x*1 also belongs to *f* (*λy.x*Ω), whereas *x ∈/ f* (nfη(*λy.x*Ω)) = *{λy.x*1*}*. Similarly, *λy.x*1[*y*] as an element of *f* (*λy.xzy*) is supposed to *η*-reduce to *x*1, while as an element of *f* (*λy.xyy*) should be *η*-normal. These examples reveal that, while the *β*-reduction of *f* (*M* ) can be performed *locally* by reducing each term individually, the *η*-reduction of nfβ*f* (*M* ) must be a *global* operation, that considers the whole set of terms before deciding whether a term should reduce or not. Rather than defining an infinitary rewriting system handling countably many terms, we prefer to divide the problem of computing the

*η*-normal form of *f* (*M* ) into two phases:

* + 1. we first define a labeling *L*(*—*) on the terms *t ∈f* (*M* ) as a global operation annotating on the empty bags 1 occurring in *t*:
* whether they “come from” a finite *η*-expansion of some variable *y*; for instance

*λy.x*1 *∈f* (*λy.x*(*λz.yz*)) should be labeled as *λy.x*1η(y),

* the set of free variables that were forgotten by taking 1 in the Taylor expansion; for instance *λy.x*1[*y*] *∈f* (*λy.xyy*) should be labeled as *λy.x*1y[*y*].
  + 1. We then define a local reduction *→*η*Æ* on *L*(nfβ*f* (*M* )) that exploits this extra-information annotated to perform the *η*-reduction only when it is safe.

The definition of the labeling *L* (Definition [5.1](#_bookmark39)) relies on a certain homogeneity exhibited by the structure of the resource terms in nfβ*f* (*M* ). As shown in [[4](#_bookmark54)], this homogeneity relies on a *deﬁnedness relation ≤* between normal resource terms:

*λx*1 *... x*n*.y ≤ λx*1 *... x*n*.y*

*t ≤ tj b ≤ bj*

*tb ≤ tjbj* 1 *≤ b*

*Etj ∈ bj 6t ∈ b, t ≤ tj b ≤ bj*

The relation *≤* is not a preorder since it is transitive, but not reflexive. For instance, *x*[*y*1[*y*]*, y*[*y*]1] */≤ x*[*y*1[*y*]*, y*[*y*]1], since *y*1[*y*] */≤ y*[*y*]1 and *y*[*y*]1 */≤ y*1[*y*]. See the discussion after Definition 9 in [[4](#_bookmark54)] for more properties of this relation, and examples. Notice that all singletons *{λx*1 *... x*n*.y}* (for *n ≥* 0) are ideals with respect to *≤*.

By Lemma 12 in [[4](#_bookmark54)], every ideal S has one of the following shapes: *{x}*, *λx.*O,

OB for some ideals O and B. Therefore, the following definition is sound.

**Definition 5.1** Let S *⊆* nfβ(Λr) be an ideal with respect to *≤* and *t ∈* S. The labeled term *L*(*t,* S) is defined as follows:

*L*(*x,{x}*)= *x, L*(*λx.t, λx.*O)= *λx.L*(*t,* O)*, L*(*tb,* OB)= *L*(*t,* O)*L*(*b,* B)*,*

⎧⎨ 1x

*L*([*t*1*,..., t*k]*,* B)= [*L*(*t*1*,* S B)*,..., L*(*t*k*,* S B)]*,* for *k >* 0

*L*(1*,* B)=

η(x)

if there exists *tj ∈* S B such that *tj* →η*′ x,* (*•*)

1fv(B) otherwise.

⎩

where *→*η*′* is *λx.t*[*x*k+1] *→*η*′ t* when *x ∈/* fv(*t*). We set *L*(S) = *{L*(*t,* S) *| t ∈* S*}*. Given a labelled term *t*, we write *t*’ for the term obtained by erasing all its labels.

The labeling *L*(*—*) can be always applied to nfβ*f* (*M* ) thanks to the following.

**Proposition 5.2** *[*[*4*](#_bookmark54)*, Lemma 23] Let M ∈* Λ*. Then* nfβ*f* (*M* ) *is an ideal w.r.t. ≤.*

**Remark 5.3** The definition of *L*(*t,* S) will be only used when S is the *β*-normal of a Taylor expansion. Under this hypothesis, the case *L*(1*,* B) is applied when B = *f* (*M* ) for some *β*-normal *M ∈* Λ and Condition (*•*) becomes “there is

S

*t ∈f* (*M* ) such that *t* →η*′ x*” which holds exactly when *M* →η *x*.

For example, for *t* = *λy.x*11 and S = nfβ*f* (*λy.x*Ω*y*) = *{λy.x*1[*y*n] *| n ≥* 0*}*

we have *L*(*t,* S) = *λy.L*(*x,{x}*)*L*(1*, {*1*}*)*L*(1*, {*[*y*k] *| k ≥* 0*}*) = *λy.x*1*$*1y . While

η(y)

1

η(y)

1

y

*L*(*λy.x*11*,* nfβ*f* (*λy.xyy*)) = *λy.x*1y

η(y)

y η(y)

. Thus *L*(*f* (*λy.xyy*)) = *{λy.x*1y

η(y)*}*

*∪{λy.x*1y

η(y)

[*y*n+1] *| n ≥* 0*}∪{λy.x*[*y*k+1]1y

*| k ≥* 0*}∪{λy.x*[*y*k+1][*y*n+1] *| n, k ≥* 0*}*.

The definition of the set f˜v(*t*) of *free variables of a labeled term t* is analogous

η(y)

˜

to the one of fv(*t*), except for the clauses f˜v(1x

)= *{x}* and f˜v(1→x)= *{→x}*.

η(x)

**Remark 5.4** Given *T* = BT(*M* ), *x ∈* fv(*T* ) iff *x ∈* fv(*t*) for every *t ∈ L*(*f* (*T* )).

**Definition 5.5** The reduction *→*η*Æ* on labelled *β*-normal resource terms, is the contextual closure of *∪*n*∈*N(*→*η*Æ* ) where *→*η*Æ* is defined as follows:

n n

(*η*l) *λx.t*1x *→ Æ t,* if *x ∈/* f˜v(*t*)*,* (*η*l ) *λx.t*[*x*n+1] *→ Æ t,* if *x ∈/* f˜v(*t*)*.*

0

η(x)

η0

n+1

ηn+1

For example, we have *L*(*λy.x*1[*y*]*,* nfβ*f* (*λy.xzy*)) = *λy.x*1z [*y*] *→*η*Æ x*1z ,

η(z)

η(z)

while *L*(*λy.x*1[*y*]*,* nfβ*f* (*λy.xyy*)) = *λy.x*1y

η(y)

*y*, which is already *η*l-normal.

**Lemma 5.6** *The reduction →*η*Æ is SN and confluent.*

**Proof** The reduction *→*η*Æ* is SN since the size of the term decreases. It is moreover weakly confluent, and therefore confluent by Newman’s lemma. *2*

**Definition 5.7** The *extensional Taylor expansion* of a *λ*-term *M* is given by:

*f* η(*M* )= nfη*Æ L*(nfβ*f* (*M* ))’

In the definition above, *β*- and *η*l-reductions are separated because the reduction

*β ∪ η*l is not confluent: for instance *λx.***I**[*x, x*] *→*η*Æ* **I** while *λx.***I**[*x, x*] *→*β *∅*.

* 1. *Eta-Reduction on B¨ohm Approximants*

We now provide the technical tools that will be used to prove Theorem [5.15](#_bookmark48). By The- orem [2.4](#_bookmark9), it is enough to prove that *f* (BTη(*M* )) is equal to nfη*Æ L*(*f* (BT(*M* )))’. The difficulty lies in that BTη(*M* ), which is the *η*-normal form of BT(*M* ), is de- fined coinductively on BT(*M* ), while the *η*l-reduction of *f* (BT(*M* )) works on a set of (labeled) resource terms coming from the finite approximants in BT(*M* )*∗*. Therefore, as an intermediate step, we define the *η*-normal form of the set BT(*M* )*∗* mimicking what we did in Subsection [5.1](#_bookmark38) for sets of resource terms. In particular, even in this framework the *η*-reduction must be a global operation; therefore, we introduce a labeling on finite approximants in the spirit of Definition [5.1](#_bookmark39).

Given M *⊆ N* , M *↓* denotes its downward closure *{a ∈ N | E b ∈* M*,a ≤⊥ b}*. When M is an ideal, we have that M = M *↓* and all its elements have a similar syntactic structure, except for *⊥*. We adopt for sets M of approximants the same syntactic sugar we used for *P*(Λr), by extending all the constructors of the grammar of *N* as pointwise operations on *P*(*N* ). For instance the ideal BT(**J***x*)*∗* can be written as *{λz*0*.x*(BT(**J***z*0)*∗*)*}↓* = *λz*0*.x*(BT(**J***z*0)*∗*) *∪ {⊥}*.

**Definition 5.8** Let M *⊆N* be an ideal w.r.t. *≤⊥* and *a ∈* M. Define *E* (*a,* M) as:

*E* (*x, {x} ↓*)= *x, E* (*λx.a,* (*λx.*M) *↓*)= *λx.E* (*a,* M *↓*)*,*

*E* (*ac,* (MN) *↓*)= *E* (*a,* M *↓*)*E* (*c,* N)*,*

⎧⎨ *⊥*

*E* (*⊥,* M)=

⎩

x η(x)

if there exists a *⊥*-free *a ∈* M such that *a* →η *x,* (*◦*)

*⊥*fv(M) otherwise*.*

We extend the definition to M by setting *E* (M)= *{E* (*a,* M) *| a ∈* M*}*. Notice that in the case (MN) *↓* above, the set N is already downward closed.

As BT(*M* )*∗* is an ideal for every *M ∈* Λ, we can always compute *L*(BT(*M* )*∗*). Condition (*◦*) is then equivalent to check that M = BT(*M j*)*∗* for some *Mj* →η *x*.

As we did for resource terms, we speak of *labeled approximants a*, we define the

set f˜v(*a*) by adding the clauses f˜v(*⊥*x ) = *{x}* and f˜v(*⊥*→x) = *{→x}*, and we write

η(x)

*a*’ for the term obtained from *a* by erasing all its labels.

**Remark 5.9** Given *T* = BT(*M* ), *x ∈* fv(*T* ) iff *x ∈* fv(*t*) for every *t ∈ E* (*T∗*).

˜

**Definition 5.10** The reduction *→*ηe on labeled approximants is defined as:

*λx.a⊥*x *→*ηe *a,* if *x ∈/* f˜v(*a*)*, λx.ax →*ηe *a,* if *x ∈/* f˜v(*a*)*.*

η(x)

It is easy to check that also *→*ηe is strongly normalizing and confluent.

After a technical lemma, we show that the *η*e-reduction on *E* (BT(*M* )) computes exactly the finite approximants of the co-inductively defined tree BTη(*M* ). Given two sets of terms X*,* Y and a reduction *→*r we write X *⇒*r Y if for all *t*1 *∈* X there is *t*2 *∈* Y such that *t*1 →r *t*2 and for all *t*2 *∈* Y there is *t*1 *∈* X such that *t*1 →r *t*2.

**Lemma 5.11** *Let T* = *λ→xy.zT*1 *··· T*k+1 *be a B¨ohm tree such that T*k+1 *is ﬁnite,*

*T*k+1 →η *y and y ∈/* fv(*zT*1 *··· T*k)*. Then E* (*T∗*) *⇒*ηe *E* ((*λ→x.zT ∗ ··· T∗*) *↓*)*.*

1 k

**Proposition 5.12** *For all M ∈* Λ*, we have* BTη(*M* )*∗* = nfηe *E* (BT(*M* )*∗*)’*.*

**Proof** [Sketch] One proceeds by co-induction on BT(*M* ) using Lemma [5.11](#_bookmark44). *2*

* 1. *A Taylor-Based Characterization of Morris’s Equivalence*

Now that the technical tools for proving the main result of the section are finally in place, we are able to prove that the extensional Taylor expansion of a *λ*-term *M* , actually captures the Taylor expansion of BTη(*M* ).

We first need the following technical results, then we show a sort of commutation between the *η*l-normalization and the Taylor expansion.

**Lemma 5.13** *Let T* = *λ→xy.zT*1 *··· T*k+1 *be a B¨ohm tree such that T*k+1 *is ﬁnite,*

*T*k+1 →η *y and y ∈/* fv(*zT*1 *··· T*k)*. Then L*(*f* (*T* )) *⇒*η*Æ L*(*f* (*λ→x.zT*1 *··· T*k))*.*

**Proposition 5.14** *For all M ∈* Λ*, f* ( nfηe *E* (BT(*M* )*∗*)’)= nfη*Æ L*(*f* (BT(*M* )))’*.*

**Proof** [Sketch] By coinduction on BT(*M* ), applying Lemma [5.13](#_bookmark46). *2*

We can finally prove the main result of the section.

**Theorem 5.15** *For every λ-term M, f* η(*M* )= *f* (BTη(*M* ))*.*

**Proof** Collecting the results above, we have the following chain of equalities:

*f* η(*M* ) = nfη*Æ L*(nfβ*f* (*M* ))’ by Definition [5.7](#_bookmark41)

= nfη*Æ L*(*f* (BT(*M* )))’ by Theorem [2.4](#_bookmark9)

= *f* ( nfηe *E* (BT(*M* )*∗*)’) by Prop. [5.14](#_bookmark47)

= *f* (BTη(*M* )*∗*) by Prop. [5.12](#_bookmark45) *2*

**Corollary 5.16** *For all M, N ∈* Λ*, we have* BTη(*M* )*∗ ⊆* BTη(*N* )*∗ if and only if*

*f* η(*M* ) *⊆f* η(*N* )*.*

**Proof** (*⇒*) Let *t ∈f* η(*M* ). Then there is *a ∈* BTη(*M* )*∗* such that *t ∈f* (*a*). Since BTη(*M* )*∗ ⊆* BTη(*N* )*∗*, we have that *a ∈* BTη(*N* )*∗*. So *t ∈f* (BTη(*N* )) and we get from Theorem [5.15](#_bookmark48) that *t ∈f* η(*N* ).

(*⇐*) Let *a ∈* BTη(*M* )*∗*. Then by Theorem [5.15](#_bookmark48) *f* (*a*) *⊆f* (BTη(*M* )) = *f* η(*M* ) *⊆ f* η(*N* ). Since *f* η(*N* ) = *f* (BTη(*N* )) holds still by Theorem [5.15](#_bookmark48), we have that *f* (*a*) *⊆f* (BTη(*N* )). From Lemma [2.3](#_bookmark8) we conclude that *a ∈* BTη(*N* )*∗*. *2*

A further corollary is that the notion of extensional Taylor expansion provides an alternative characterization of Morris’s equivalence.

**Corollary 5.17** *For M, N ∈* Λ*, we have M ≡*nf *N if and only if f* η(*M* )= *f* η(*N* )*.*

**Proof** We have the following chain of equivalences: By [[25](#_bookmark75)] *M ≡*nf *N* if and only if BTη(*M* )= BTη(*N* ), that is BTη(*M* )*∗* = BTη(*N* )*∗*. By Corollary [5.16](#_bookmark49) this holds if and only if *f* η(*M* )= *f* η(*N* ) does. *2*

# Related and Further Works

In [[7](#_bookmark57)], Coppo, Dezani and Zacchi defined a filter model *D*cdz having two non-trivially ordered atoms *ϕT ≤ ϕ*٨ and proved that its theory is *H*+, namely the theory of Morris’s equivalence. We claim that the relational semantics provides a more natural framework for building models having this theory since: (*i*) it is enough for an rgm to preserve *ω*-polarities to induce *H*+ as equational theory; (*ii*) the resulting models are simpler than filter models as their elements are trivially ordered; (*iii*) a range of powerful tools coming from the resource calculus, like the Taylor expansion, are available as the relational semantics is actually a model of differential linear logic. The present article is reminiscent of [[16](#_bookmark67)], where the first author gives sufficient conditions for models living in non-well-pointed categories (in particular the re- lational semantics) to have as theory *H∗*. The proof techniques used in [[16](#_bookmark67)] are however more standard as those categories are not necessarily models of differential linear logic. A breakthrough in this subject has been recently achieved by Breuvart in [[5](#_bookmark55)], where he was able to provide a precise characterisation of those Krivine’s models (*K*-models, for short) having theory *H∗*. Indeed he proved that an exten- sional *K*-model has theory *H∗* if and only if the unfolding of equivalent arrow types is governed by a *hyperimmune* function, a notion widely used in recursion theory. It would be interesting to check whether an analogous result holds for rgms, and to look for necessary and sufficient conditions for characterising those rgms having

theory *H*+.

Concerning the syntactic results presented in Section [5](#_bookmark37), it would be interesting to look for a notion of extensional Taylor expansion capturing directly Morris’s pre- order. This would allow to strengthen Theorem [4.5](#_bookmark32) by adding a further syntactic characterisation of *±*nf based on Taylor approximants.

A more ambitious goal is to generalise the definition of extensional Taylor ex- pansion to the full fragment of resource calculus, where the notion of B¨ohm tree has no easy equivalent. Preliminary investigations by the first author and Pagani [[18](#_bookmark68)] show that a B¨ohm-like theorem holds in that setting: two finite sums of normal resource terms (possibly with promotion) are semi-separable with respect to may- convergence to a head-normal forms exactly when they are not *η*-convertible or Taylor-equivalent. The problems of fully characterising observational equivalences with respect to normal form or head normal forms for sums of arbitrary resource terms either semantically (in terms of relational models) or syntactically (in terms of Taylor expansion) are still open and promise to be quite difficult.

# Acknowledgement

Thanks to Henk Barendregt, Mariangiola Dezani, Thomas Ehrhard, Stefano Guer- rini, Michele Pagani for stimulating discussions and the reviewers for their com- ments.

# References

1. Amadio, R. and P.-L. Curien, “Domains and Lambda Calculi,” Cambridge tracts in theoretical computer science, Cambridge University Press, 1998.
2. Barendregt, H., “The lambda-calculus, its syntax and semantics,” Number 103 in Stud. Logic Found. Math., North-Holland, 1984, second edition.
3. Berline, C., *From computation to foundations via functions and application: The* λ*-calculus and its* *webbed models*, Theor. Comput. Sci. **249** (2000), pp. 81–161.
4. Boudes, P., F. He and M. Pagani, *A characterization of the Taylor expansion of lambda-terms*, in:

S. Ronchi Della Rocca, editor, *CSL’13*, LIPIcs **23** (2013), pp. 101–115.

1. Breuvart, F., *On the characterization of models of Hs* (2014), To appear in Computer Science Logic and Logic in Computer Science (CSL-LICS 2014).
2. Bucciarelli, A., T. Ehrhard and G. Manzonetto, *Not enough points is enough*, in: *CSL’07*, LNCS **4646**

(2007), pp. 298–312.

1. Coppo, M., M. Dezani and M. Zacchi, *Type theories, normal forms and* D*∞-lambda-models*, Inf. Comput. **72** (1987), pp. 85–116.
2. Ehrhard, T. and L. Regnier, *The differential lambda-calculus*, Theor. Comput. Sci. **309** (2003), pp. 1–41.
3. Ehrhard, T. and L. Regnier, *B¨ohm trees, Krivine’s machine and the Taylor expansion of lambda-terms*, in: *CiE*, LNCS **3988**, 2006, pp. 186–197.
4. Ehrhard, T. and L. Regnier, *Uniformity and the Taylor expansion of ordinary lambda-terms*, Theor. Comput. Sci. **403** (2008), pp. 347–372.
5. Engeler, E., *Algebras and combinators*, Algebra Universalis **13** (1981), pp. 389–392.
6. Hyland, J., *A survey of some useful partial order relations on terms of the lambda calculus*, in: C. B¨ohm, editor, *Lambda-Calculus and Computer Science Theory*, LNCS **37** (1975), pp. 83–95.
7. Hyland, J., *A syntactic characterization of the equality in some models for the* λ*-calculus*, J. London Math. Soc. (2) **12(3)** (1975/76), pp. 361–370.
8. Hyland, M., M. Nagayama, J. Power and G. Rosolini, *A category theoretic formulation for Engeler-style* *models of the untyped* λ*-calculus*, Electronic Notes in Theor. Comp. Sci. **161** (2006), pp. 43–57.
9. Levy, J.-J., *Le lambda calcul - notes du cours* (2005), in French, [http://pauillac.inria.fr/~levy/](http://pauillac.inria.fr/~levy/courses/X/M1/lambda/dea-spp/jjl.pdf) [courses/X/M1/lambda/dea-spp/jjl.pdf](http://pauillac.inria.fr/~levy/courses/X/M1/lambda/dea-spp/jjl.pdf).
10. Manzonetto, G., *A general class of models of Hs*, in: *MFCS 2009*, LNCS **5734** (2009), pp. 574–586.
11. Manzonetto, G., *What is a categorical model of the differential and the resource* λ*-calculi?*, Mathematical Structures in Computer Science **22** (2012), pp. 451–520.
12. Manzonetto, G. and M. Pagani, *B¨ohm’s theorem for resource lambda calculus through taylor expansion*, in: *TLCA 2011*, LNCS **6690**, 2011, pp. 153–168.
13. Morris, J., “Lambda calculus models of programming languages,” Ph.D. thesis, MIT (1968).
14. Nakajima, R., *Infinite normal forms for the lambda calculus*, in: *Lambda-Calculus and Computer Science Theory*, Lecture Notes in Computer Science **37** (1975), pp. 62–82.
15. Paolini, L., M. Piccolo and S. Ronchi Della Rocca, *Logical relational* λ*-models* (2014), draft available at <http://www.di.unito.it/~paolini/papers/logicalRelational.pdf>.
16. Ronchi Della Rocca, S. and L. Paolini, “The Parametric λ-Calculus: a Metamodel for Computation,” Texts in TCS: An EATCS Series, Springer-Verlag, Berlin, 2004.
17. Scott, D., *Continuous lattices*, in: Lawvere, editor, *Toposes, Algebraic Geometry and Logic*, Lecture Notes in Math. **274** (1972), pp. 97–136.
18. Tranquilli, P., *Intuitionistic differential nets and* λ*-calculus*, Th. Comp. Sci. **412** (2011), pp. 1979–1997.
19. van Bakel, S., F. Barbanera, M. Dezani-Ciancaglini and F.-J. de Vries, *Intersection types for lambda- trees*, Theor. Comput. Sci. **272** (2002), pp. 3–40.

# Technical Appendix

This technical appendix is devoted to provide some proofs that were omitted or just sketched in the article.

* 1. *Omitted proofs of Section* [*2*](#_bookmark4)

**Lemma 2.3** *Let a ∈N and M ∈* Λ*, then f* (*a*) *⊆f* (BT(*M* )) *entails a ∈* BT(*M* )*∗.*

**Proof** By structural induction on *a*.

**Case** *a* = *⊥* and *f* (*a*)= *∅⊆f* (BT(*M* )). Then it is trivial since *⊥∈* BT(*M* ).

**Case** *a* = *λ→x.ya*1 *··· a*k and *f* (*a*) = Sn ,...,n *≥*0 *λ→x.y*[*f* (*a*1)n1 ] *···* [*f* (*a*k)n*k* ] *⊆*

1

*k*

*f* (BT(*M* )). Then *M* →β *λ→x.yN*1 *··· N*k for some *N*1*,..., N*k *∈* Λ such that *f* (*a*i) *⊆*

*f* (BT(*N*i)). By induction hypothesis *a*i *∈* BT(*N*i)*∗* for all 1 *≤ i ≤ k* and we conclude that *λ→x.ya*1 *··· a*k *∈* BT(*M* )*∗*. *2*

* 1. *Omitted proofs of Section* [*3*](#_bookmark10)

**Theorem 3.10 (Approximation Theorem)** *Let M be a λ-term. Then* Γ *▶ M* :

*σ if and only if there exists t ∈f* (*M* ) *such that* Γ *▶ t* : *σ.*

**Proof** (*⇒*) The proof is by induction on a derivation of Γ *▶ M* : *σ*. We proceed by case analysis on the last rule applied in the derivation.

**Case** var**.** We have *x* : *σ ▶ x* : *σ* using the rule (var). This case is trivial since

*f* (*x*)= *{x}*.

**Case** lam**.** We have Γ *▶ λx.N* : *σ* using the rule (lam). By Theorem [3.7](#_bookmark15)([ii](#_bookmark16)), we have that Γ*,x* : *μ ▶ N* : *τ* for some *μ → τ σ*. By IH, there exists *tj ∈f* (*N* ) such that Γ*,x* : *μ ▶ tj* : *τ* . Therefore, *λx.tj ∈f* (*λx.N* ) and

Γ*,x* : *μ ▶ tj* : *τ*

Γ *▶ λx.tj* : *μ → τ* (lam) *μ → τ σ*

Γ *▶ λx.tj* : *σ* (eq)

**Case** app**.** We have Γ *▶ NP* : *σ* using the rule (app). By Theorem [3.7](#_bookmark15)([iii](#_bookmark18)), there

is a decomposition Γ = Γ0 *∧* (*∧*n

i=1

Γi) for some *n ≥* 0, such that Γ0 *▶ N* : *∧*n

*σ*i *→ σ*

and Γi *▶ P* : *σ*i. By IH, there exists *s ∈f* (*N* ) such that Γ0 *▶ s* : *∧*n

i=1

i=1

*σ*i *→ σ*, and

there exist *t*1*,..., t*n *∈f* (*P* ) such that Γi *▶ t*i : *σ*i.

Therefore we have that *s*[*t*1*,..., t*n] *∈f* (*NP* ) and:

Γ0 *▶ s* : *∧*n *σ*i *→ σ* Γi *▶ t*i : *σ*i *6i ∈ {*1*,..., n}*

i=1

Γ *▶ s*[*t*1*,..., t*n]: *σ*

**Case** eq**.** Let Γ *▶ M* : *σ* using the rule (eq). Then Γ *▶ M* : *τ* for some *τ σ*. By IH there exists *t ∈ f* (*M* ) such that Γ *▶ t* : *τ* . By applying (eq) we derive Γ *▶ t* : *σ*.

This concludes the left-to-right implication.

(*⇐*) Let *t ∈f* (*M* ) such that Γ *▶ t* : *σ*. We proceed by induction on the deriva- tion for such a type assignment.

**Case** var**.** We have *x* : *σ ▶ x* : *σ* and *x ∈ f* (*M* ) which entails *M* = *x* by definition of the Taylor expansion. This case is therefore trivial.

**Case** app**.** We have *s*[*t*1*,..., t*n] *∈ f* (*M* ) such that Γ *▶ s*[*t*1*,..., t*n] : *σ*. By

Theorem [3.7](#_bookmark15)([iii](#_bookmark18))’, we get the decomposition Γ = Γ0 *∧* (*∧*n

i=1

Γi) and the typing

assignments Γ0 *▶ t* : *∧*n

i=1

*σ*i *→ σ* and Γi *▶ s*i : *σ*i. By definition of Taylor expansion,

if *s*[*t*1*,..., t*n] *∈f* (*M* ) then *M* = *NP* for some *N, P ∈* Λ such that *s ∈f* (*N* ) and

*t*1*,..., t*n *∈f* (*P* ). By IH, Γ0 *▶ N* : *∧*n

i=1

*σ*i *→ σ* and Γi *▶ P* : *σ*i for all *i ∈ {*1*,..., n}*.

Therefore we derive:

Γ0 *▶ N* : *∧*n *σ*i *→ σ* Γi *▶ P* : *σ*i *6i ∈ {*1*,..., n}*

i=1 (app)

Γ *▶ NP* : *σ*

**Case** lam**.** We have *λx.t ∈f* (*M* ) such that Γ *▶ λx.t* : *σ*. By definition of Taylor expansion, *λx.t ∈ f* (*M* ) entails *M* = *λx.N* for some *N ∈* Λ such that *t ∈ f* (*N* ). By Theorem [3.7](#_bookmark15)([ii](#_bookmark16)), one gets Γ*,x* : *μ ▶ t* : *τ* for some *μ → τ σ*. By IH, we have Γ*,x* : *μ ▶ N* : *τ* . Therefore, we can derive

Γ*,x* : *μ ▶ N* : *τ*

Γ *▶ λx.N* : *μ → τ* (lam) *μ → τ σ*

Γ *▶ λx.N* : *σ* (eq)

**Case** eq**.** Let *t ∈f* (*M* ) and suppose Γ *▶ t* : *σ* comes from Γ *▶ t* : *τ* by (eq). By IH, we have Γ *▶ M* : *τ* . By applying (eq) we derive Γ *▶ N* : *σ*. *2*

* 1. *Omitted proofs of Section* [*4*](#_bookmark23)

We recall the definition of “*ω* occurs positively/negatively in a type *σ*”.

**Definition A.1** The relations *ω ∈*p *σ* for *p ∈ {*+*, —}* are defined as follows:

* + 1. *ω ∈— μ → σ* for any type *σ* and intersection *μ* such that *μ* = *ω*;
    2. if *ω ∈*p *σ* then *ω ∈*p *μ → σ* for any intersection *μ*;
    3. if *ω ∈*p *σ* then *ω ∈ч*p *σ ∧ μ → τ* for any types *σ, τ* and intersection *μ*.

Remark that the condition *μ* = *ω* in ([i](#_bookmark77)) is non-trivial, since equality = between types includes the neutrality of *ω*. For instance *ω ∈— ω ∧ ω → σ* as *ω ∧ ω* = *ω*.

**Proposition 4.2** *Let A be a partial pair such that, for all m ∈ M*f(*A*) *and α ∈ A,*

(*m, α*) *∈* dom(*j*) *entails that m /*= []*. Then A preserves ω-polarities.*

**Proof** We perform an induction loading and prove that, for all type *σ, τ ∈* T*A* and *p ∈ {*+*, —}*: if *ω ∈*p *σ* and *τ A σ* then *τ• ∈/ A* and *ω ∈*p *τ* . In the rest of the proof we will just writefor *A*.

We proceed by induction on the definition of *ω ∈*p *σ*.

**Case (**[**i**](#_bookmark77)**).** Suppose that *ω ∈*p *σ* because *p* = *—* and *σ* = *ω → γ*, then we need to prove that *ω ∈— τ* , for any *τ* such that *τ σ*, that is such that *τ•* = *σ•*. By

definition, we have:

*σ•* = (*ω → γ*)*•* = *j*([]*, γ•*)= ([]*, γ•*)

where the last equality follows from Definition [3.2](#_bookmark11) and the hypothesis that ([]*, γ•*) *∈/*

dom(*j*). From *τ•* = ([]*, γ•*) we get that *τ• ∈/ A* since *A* does not contain any pair, and this entails that also *τ* cannot be atomic.

Suppose therefore *τ* = *μ → δ*, then we have *τ•* = (*μ → δ*)*•* = *j*(*μ•, δ•*) = *j*([]*, γ•*) = *σ•.* From the injectivity of *j*, we get that *μ•* = [] and *δ•* = *γ•*, so *τ* = *ω → δ* and *ω ∈— τ* .

**Case (**[**ii**](#_bookmark78)**).** Suppose that *ω ∈*p *σ* because *σ* = *μ → γ* and *ω ∈*p *γ*. Then

*σ•* = (*μ → γ*)*•* = *j*(*μ•, γ•*)= *τ•.*

From *ω ∈*p *γ*, *γ γ* and the induction hypothesis, we get that *γ• ∈/ A* and therefore (*μ•, γ•*) *∈/* dom(*j*). By Definition [3.2](#_bookmark11), we have that *j*(*μ•, γ•*) = (*μ•, γ•*), and since this is equal to *τ•*, we get *τ* = *ν → δ* for some *ν, δ*. From *j*(*μ•, γ•*)= *j*(*ν•, δ•*) and the injectivity of *j* we get that *μ•* = *ν•* and *γ•* = *δ•*.

From *ω ∈*p *γ* and *δ γ* we get, by induction hypothesis, that *ω ∈*p *δ* and therefore *ω ∈*p *ν → δ* = *τ* .

**Case (**[**iii**](#_bookmark79)**).** Suppose that *ω ∈*p *σ* because *σ* = *γ*1 *∧ μ → γ*2 and *ω ∈ч*p *γ*1. From

*γ*1 *∧ μ → γ*2 *τ* , we get

(*γ*1 *∧ μ → γ*2)*•* = *j*([*γ•*]+ *μ•, γ•*)= *τ•.*

1 2

Suppose, by the way of contradiction, that *τ* is an atomic type *α*. Then, we have

*j*([*γ•*]+ *μ•, γ•*)= *α* which implies, by Definition [3.2](#_bookmark11), that ([*γ•*]+ *μ•, γ•*) *∈* dom(*j*) *⊆*

1 2 1 2

*M*f(*A*) *× A*. In particular, we get *γ• ∈ A*, which is impossible since *ω ∈ч*p *γ*1 and

1

*γ*1 *γ*1, so by the induction hypothesis we conclude that *γ•* is not atomic.

1

So, *τ* = *ν → δ*2, and (*ν → δ*2)*•* = *j*(*ν•, δ•*)= *j*([*γ•*]+ *μ•, γ•*)*.* Since *j* is injective,

2 1 2

*ν•* = [*γ•*]+ *μ•* and *δ•* = *γ•*. Therefore, *ν* = *δ*1 *∧ νj* such that *δ•* = *γ•* and *νj•* = *μ•*.

1 2 2 1 1

Since *ω ∈ч*p *γ*1 and *γ*1 *δ*1, by IH we get *ω ∈ч*p *γ*1 and we conclude that *ω ∈*p *τ* . *2*

For convenience, we present Lemma [4.3](#_bookmark26) with an additional equivalent sen- tence ([iii](#_bookmark80)-bis), which is an intermediate step between ([iii](#_bookmark29)) and ([iv](#_bookmark30)).

**Lemma 4.3** *Let M ∈* Λ*. The following are equivalent:*

1. *M has a normal form,*
2. *there is a ∈* BT(*M* )*∗ that does not contain ⊥,*
3. *there is t ∈* nfβ(*f* (*M* )) *that does not contain the empty bag* 1*,*

(iii-bis) *in every rgm D preserving ω-polarities,* Γ *▶Q t* : *σ for some t ∈* nfβ*f* (*M* )*, environment* Γ *and type σ such that ω ∈/*+ *σ and ω ∈/—* Γ*, that is ω ∈/—* Γ(*x*) *for all x ∈* Var*.*

1. *in every rgm D preserving ω-polarities,* Γ *▶Q M* : *σ for some environment* Γ

*and type σ such that ω ∈/*+ *σ and ω ∈/—* Γ*, that is ω ∈/—* Γ(*x*) *for all x ∈* Var*.*

**Proof** ([*i*](#_bookmark27) *⇐⇒* [*ii*](#_bookmark28)) is trivial.

([*ii*](#_bookmark28) *⇐⇒* [*iii*](#_bookmark29)) follows from Theorem [2.4](#_bookmark9).

*(*[*iii*](#_bookmark29) *⇒* [*iii*](#_bookmark80)*-bis)* We prove that this implication holds more generally for any *β*- normal form *t* that does not contain 1 (regardless the fact that *t* belongs to a Taylor expansion). We proceed by structural induction on *t*.

**Case** *t* = *λx.tj* where *tj* is *β*-normal. By induction hypothesis, Γ*j ▶ tj* : *τ* holds for some context Γ*j* and type *τ* such that *ω ∈/—* Γ*j* and *ω ∈/*+ *τ* . Note that Γ*j* can be written as Γ*,x* : *μ* for some Γ and *μ*, therefore we can derive:

Γ*,x* : *μ ▶ tj* : *τ*

Γ *▶ λx.tj* : *μ → τ* (lam)

From *ω ∈/—* Γ*j* we get that *ω ∈/—* Γ and *ω ∈/— μ*, which entails *ω ∈/*+ *μ → τ* .

**Case** *t* = *yb*1 *··· b*k, for some *k ≥* 0, and each *b*i = [*s*i,1*,..., s*i,n*i* ] (for *n*i *≥* 0) only contains *β*-normal terms. By induction hypothesis, there are environments Γij, and types *τ*ij, such that *ω ∈/—* Γij and *ω ∈/*+ *τ*ij and Γij *▶ s*ij : *τ*ij holds. Then we can derive:

Γ0 *▶ y* : *μ*1 *→ · · · → μ*k *→ α* Γij *▶ s*ij : *τ*ij *i ∈ {*1*,..., k}, j ∈ {*1*,..., n*i*}*

i=1

Γ *▶ yb*1 *··· b*k : *α*

where *μ*i = *∧*n*i*

j=1

*τ*ij, Γ0 = *y* : *μ*1 *→ ··· → μ*k *→ α* and Γ = Γ0 *∧* (*∧*k

n*i* j=1

Γij). As

*ω ∈/*+ *τ*ij we also have *ω ∈/— μ*i and therefore *ω ∈/—* Γ0. From this, and the hypotheses that *ω ∈/—* Γij we get that *ω ∈/—* Γ. Of course *ω ∈/— α* because *α* is an atom.

*∧*

*(*[*iii*](#_bookmark80)*-bis ⇒* [*iii*](#_bookmark29)*)* Consider *t ∈* nfβ*f* (*M* ) such that Γ *▶ t* : *σ* where Γ and *σ* satisfy the hypotheses of ([iii](#_bookmark80)-bis). We proceed by induction on the structure of the *β*- normal *t*.

**Case** *t* = *λx.tj* where *tj* is *β*-normal. By applying Theorem [3.7](#_bookmark15)([ii](#_bookmark16)) we have that Γ*,x* : *μ ▶ tj* : *τ* holds for *μ ∈* I*Q* and *τ ∈* T*Q* such that *σ μ → τ* . Since

*D* preserves *ω*-polarities, *ω*

*∈/*+ *σ* entails *ω*

*∈/*+ *μ → τ* . As neither Γ nor *μ* has

negative occurrences of *ω*, we have *ω ∈/—* (Γ*,x* : *μ*) and *ω ∈/*+ *τ* , so, by the induction hypothesis, we get that *tj* does not have occurrences of 1. Therefore 1 does not occur in *λx.tj* either.

**Case** *t* = *yb*1 *··· b*k, for some *k ≥* 0, and each *b*i = [*s*i,1*,..., s*i,n*i* ] (for *n*i *≥* 0) only contains *β*-normal terms. If *k* = 0 we are done, as *y* does not contain 1. Consider then the case *k >* 0. By iterating Theorem [3.7](#_bookmark15)([iii](#_bookmark18)’) we know that there is

a decomposition Γ = Γ0 *∧* (*∧*k

i=1

*∧*

j=1

n*i* j=1

Γij) such that (setting *μ*i = *∧*n*i*

*τ*ij):

Γ0 *▶ y* : *μ*1 *→ · · · → μ*k *→ σ* Γij *▶ s*ij : *τ*ij for *i* = 1*,...,k j* = 1*,..., n*i

Γ *▶ yb*1 *··· b*k : *σ*

By Theorem [3.7](#_bookmark15)([i](#_bookmark17)), we get that Γ0 = *y* : *τ* for some *τ μ*1 *→ · · · → μ*k *→ σ*. From this, it follows that Γ(*y*)= *τ ∧ μ* for some *μ*, so *ω ∈/—* Γ entails that *ω ∈/— τ* and, as *D* preserves *ω*-polarities, we get that *ω ∈/— μ*1 *→ ··· → μ*k *→ σ*. From this, on the one side we get that each *μ*i is different from *ω* (that is, *n*i *>* 0, so *b*i */*= 1) and on the other side that *ω ∈/*+ *τ*ij holds for 1 *≤ i ≤ k* and 1 *≤ j ≤ n*i. We can therefore

apply the induction hypothesis to each derivation Γij *▶ s*ij : *τ*ij and conclude that the terms *s*ij do not contain 1, so neither does the term *yb*1 *··· b*k.

([*iii*](#_bookmark80)*-bis ⇐⇒* [*iv*](#_bookmark30)) Let us suppose ([iv](#_bookmark30)). By Theorem [3.10](#_bookmark76), we have thatΓ *▶ M* : *σ* holds if and only if there exists *s ∈ f* (*M* ) such that Γ *▶ s* : *σ*. By Theorem [2.1](#_bookmark6) (strong normalization of Λr) and Theorem [3.8](#_bookmark19)([i](#_bookmark20))-([ii](#_bookmark21)) (both subject reduction and expansion), the latter is equivalent to the existence of *t ∈* nfβ*f* (*M* ) such that Γ *▶ t* : *σ*. Therefore, ([iv](#_bookmark30)) is equivalent to ([iii](#_bookmark80)-bis). *2*

For proving Lemma [4.4](#_bookmark31), we need the following remark and technical lemma.

**Remark A.2** In the model *D*٨, we have that *σ* holds if and only if *σ* is generated by the following grammar:

*γ* ::= *| γ → γ*

In particular, *μ → σ* entails that *μ* = *τ* for some *τ* and *σ* .

**Lemma A.3** *Let N ∈* Λ *be a β-normal form. If* Γ *▶ N* : *σ, for some* Γ *and σ such that* Γ(*x*) *for all x ∈* dom(Γ) *and σ , then N is linear and* dom(Γ) = fv(*N* )*.*

**Proof** We proceed by structural induction on *N* .

**Case** *N* = *λx.Nj* where *Nj* is *β*-normal. From Γ *▶ λx.Nj* : *σ* we get, by Theorem [3.7](#_bookmark15)([ii](#_bookmark16)), that Γ*,x* : *μ ▶ Nj* : *τ* for some *μ, τ* such that *μ → τ σ* and, by transitivity of, we get that *μ → τ* holds. By Remark [A.2](#_bookmark81) this entails *μ* = *γ* for some *γ* and *τ* , therefore we can apply the induction hypothesis and get that *Nj* is linear and dom(Γ*,x* : *γ*) = fv(*Nj*). Thus, *λx.Nj* is also linear and dom(Γ) = fv(*Nj*) *— {x}* = fv(*λx.Nj*) which is what we are meant to prove.

**Case** *N* = *yN*1 *··· N*k such that *N*1*,..., N*k are *β*-normal. By Theorem [3.7](#_bookmark15)([iii](#_bookmark18))

*∧*

and there is a decomposition Γ = Γ0 *∧* (*∧*k

i=1

n*i* j=1

Γij) such that Γ0 *▶ y* : *μ*1 *→*

*··· → μ*k *→ σ* holds for some *μ*i = *τ*i1 *∧ · · ·∧ τ*in*i* and Γij *▶ N*i : *τ*ij is derivable

for all 1 *≤ i ≤ k* and 1 *≤ j ≤ n*i. By Theorem [3.7](#_bookmark15)([i](#_bookmark17)), Γ0 = *y* : *γ*, for a type *γ μ*1 *→ · · · → μ*k *→ σ*. As Γ0(*y*) = Γ(*y*) = *γ* we also have by transitivity ofthat *μ*1 *→ ··· → μ*k *→ σ* which entails by Remark [A.2](#_bookmark81) that *μ*i = *τ*i

(i.e. *n*i = 1) and *τ*i. Therefore we have Γ = Γ0 *∧* (*∧*k

i=1

Γi) and Γi *▶ N*i : *τ*i for

some Γi such that Γi(*x*)for all *x ∈* dom(Γi) and *τ*i.

By the induction hypothesis we get that each *N*i is linear and dom(Γi)= fv(*N*i).

We conclude that *yN*1 *··· N*k is linear and dom(Γ) = dom(Γ0) *∪* (Sk dom(Γi)) =

i=1

fv(*yN*1 *··· N*k). *2*

**Lemma 4.4** *Let M ∈* Λ *and* Γ= *x*1 : *,..., x*n : *. Then* Γ *▶Qs M* : *if and only if M has a linear β-normal form and* dom(Γ) = fv(nfβ(*M* ))*.*

**Proof** (*⇒*) By Theorem [4.2](#_bookmark25), the rgm *D*٨ preserves *ω*-polarities. As *ω* does not occur positively nor negatively in, we can deduce by Lemma [4.3](#_bookmark26) that *M* has a *β*- normal form. By subject reduction, we derive Γ *▶* nfβ(*M* ):and, by Lemma [A.3](#_bookmark82), we conclude that nfβ(*M* ) is linear.

(*⇐*) Suppose that *M ∈* Λ has a linear *β*-normal form and that the environ- ment Γ = *x*1 : *,..., x*n :is such that dom(Γ) = fv(nfβ(*M* )). It is enough to

prove that Γ *▶* nfβ(*M* ) :is derivable, then one concludes by subject expansion (Theorem [3.8](#_bookmark19)([i](#_bookmark20))) that Γ *▶ M* :holds. We proceed by induction on nfβ(*M* ).

**Case** nfβ(*M* ) = *λx.Nj* where *Nj* is *β*-normal. Obviously, *Nj* is linear and dom(Γ*,x* :)= fv(*N j*), so by the induction hypothesis

Γ*,x* : *▶ Nj* :

Γ *▶ λx.Nj* : *→* (lam)  *→*

Γ *▶ λx.Nj* :(eq)

is also derivable.

**Case** nfβ(*M* )= *yN*1 *··· N*k such that *N*1*,..., N*k are *β*-normal. We let Γi to be the environment such that Γi(*x*) =if *x ∈* fv(*N*i) and Γi(*x*) = *ω* otherwise. As the *N*i’s are linear, we derive Γi *▶ N*i :by the induction hypothesis. Then we can derive (for Γ0 = *y* :):

Γ0 *▶ y* :(var)  *→ · · · → →*

Γ0 *▶ y* : *→ · · · → →* (eq) Γi *▶ N*i :1 *≤ i ≤ k*

Γ *∧* (*∧*k Γ ) *▶ yN ··· N*

(lam)

:

0 i=1 i 1 k

To conclude, it is enough to check that Γ = Γ0 *∧* (*∧*k

i=1

Γi). *2*

*A.4 Omitted proofs of Section* [*5*](#_bookmark37)

**Lemma A.4** *Let M ∈* Λ *be a β-normal form such that M* →η *x. For all a ∈ M∗,*

*we have that either E* (*a, M∗*)= *⊥*x *or E* (*a, M∗*) →ηe *x.*

η(x)

**Proof** Since *M* is *β*-normal, it has the shape *λx*1 *... x*n*.xN*1 *··· N*k. As *M* →η *x*

we get that *n* = *m*, *x /*= *x*i and *N*i →β *x*i for all *i ∈ {*1*,..., n}*.

We proceed by induction on *a*.

**Case** *a* = *⊥*. Then *E* (*a, M∗*)= *⊥*x

η(x)

by Definition [5.8](#_bookmark42).

**Case** *a* = *λx*1 *... x*n*.xa*1 *··· a*n with *a*i *∈ N∗* for all *i ∈ {*1*,..., n}*. By induc-

i

*∗ ∗* x*i ∗*

tion hypothesis, either *E* (*a*i*, N*i ) →ηe *x*i or *E* (*a*i*, N*i ) →ηe *⊥*η(x*i*) so *E* (*a, M* ) =

*λx*1 *... x*n*.xE* (*a*1*,N∗*) *···E* (*a*n*,N∗*) →ηe *x*. *2*

1 n

**Lemma 5.11** *Let T* = *λ→xy.zT*1 *··· T*k+1 *be a B¨ohm tree such that T*k+1 *is ﬁnite,*

*T*k+1 →η *y and y ∈/* fv(*zT*1 *··· T*k)*. Then E* (*T∗*) *⇒*ηe *E* ((*λ→x.zT ∗ ··· T∗*) *↓*)*.*

1 k

**Proof** We first prove that, given *a ∈ T∗*, there exists *aj ∈* (*λ→x.zT ∗ ··· T∗*) *↓* such

1 k

that *E* (*a, T∗*) →ηe *E* (*aj,* (*λ→x.zT ∗ ··· T∗*) *↓*). We split into cases depending on *a*.

1 k

**Case** *a* = *⊥*. Then *E* (*a, T∗*) = *E* (*⊥,T∗*) = *E* (*⊥,* (*λ→xy.zT ∗ ··· T∗T∗*

) *↓*). From

1 k k+1

the fact that *T*k+1 is finite, we get that *T*k+1 *∈N* and since *T*k+1 →η *y* we have that

*T*k+1 is *⊥*-free. As *y ∈/* fv(*zT*1 *··· T*k), there is a *⊥*-free *c*1 *∈ T∗* such that *c*1 →η *z* if and only if there exists a *⊥*-free *c*2 *∈* (*λ→x.zT ∗ ··· T∗*) *↓* such that *c*2 →η *z*. Therefore

1 k

*E* (*⊥,* (*λ→xy.zT ∗ ··· T∗T∗* ) *↓*)= *E* (*⊥,* (*λ→x.zT ∗ ··· T∗*) *↓*), so *aj* = *⊥*.

1 k k+1 1 k

**Case** *a* = *λ→xy.za*1 *··· a*k+1, with *a*i *∈ T∗*

i

for 1 *≤ i ≤ k* + 1. By defini-

tion, we have *E* (*a, T∗*) = *λ→xy.zE* (*a*1*,T∗*) *···E* (*a*k*,T∗*)*E* (*a*k+1*,T∗*

). By hypothe-

1 k k+1

sis, *T*k+1 is actually a *λ*-term (i.e., finite and *⊥*-free) such that *T*k+1 →η *y* so, by Lemma [5.11](#_bookmark44), either *E* (*a*k+1*, T*k+1) →ηe *⊥*η(y) or *E* (*a*k+1*, T*k+1) →ηe *y*. By Remark [5.9](#_bookmark43) *y ∈/* fv(*zT*1 *··· T*k) entails *y ∈/* f˜v(*zE* (*a*1*,T∗*) *···E* (*a*k*,T∗*)), hence in both cases we get

y

1

k

*E* (*a, T∗*) →ηe *λ→x.zE* (*a*1*,T∗*) *···E* (*a*k*,T∗*) *∈ E* ((*λ→x.zT ∗ ··· T∗*) *↓*). Therefore the *aJ* we

1 k 1 k

were looking for is just *λ→x.za*1 *··· a*k.

Second, we prove that for every *aJ ∈* (*λ→x.zT ∗ ··· T∗*) *↓* there is *a ∈ T∗* such that

1 k

*E* (*a, T∗*) →ηe *E* (*aJ,* (*λ→x.zT ∗ ··· T∗*) *↓*). Again, we split into cases depending on *aJ*.

1 k

**Case** *aJ* = *⊥*. It is enough to take *aJ* = *⊥* and reason as above.

**Case** *aJ* = *λ→x.zaJ ··· aJ* with *aJ ∈ T∗* for all 1 *≤ i ≤ k*. Clearly, *⊥ ∈ T∗*

and

1 k i i

*∗* y

k+1

*E* (*⊥, T*k+1) = *⊥*η(y), since by hypothesis *T*k+1 is finite and *T*k+1 →η *y*. Therefore,

for *a* = *λ→xy.zaJ ··· aJ ⊥∈ T∗* we have

1 k

*E* (*a, T∗*) = *λ→xy.zE* (*aJ ,T∗*) *···E* (*aJ ,T∗*)*E* (*aJ ,T∗* )

1 1 k k

k+1

k+1

= *λ→xy.zE* (*aJ ,T∗*) *···E* (*aJ ,T∗*)*⊥*y

1 1 k k η(y)

*→*η*e λ→x.zE* (*aJ ,T∗*) *···E* (*aJ ,T∗*) using Remark [5.9](#_bookmark43)

1 1 k k

= *E* (*aJ, λ→x.zT ∗ ··· T∗*)*.*

1 k

We conclude as *E* (*a, T∗*) *∈ E* (*T∗*). *2*

**Lemma A.5** *For all B¨ohm trees T, we have η*(*T* )*∗* = nfηe (*E* (*T∗*))’*.*

**Proof** We proceed by co-induction on *T* .

If *T* = *⊥*, then *η*(*T* )*∗* = *{⊥}* = *{* *⊥$*’*}* = *{* *E* (*⊥, ⊥*)’*}* = nfηe (*E* (*T∗*))’. Otherwise, the B¨ohm tree *T* can be written in a unique way as *T* =

*λx*1 *... x*n*y*1 *... y*m*.zT*1 *··· T*k*T J ··· T J*

(for some *n, m, k ≥* 0) such that:

1 m

* *y*i *∈/* fv(*zT*1 *··· T*k), *T J* is finite and *T J* →η *y*i for all *i ∈ {*1*,..., m}*,

i i

* *x*n *∈* fv(*zT*1 *··· T*k) or *T*k is infinite, or *T*k is finite but does not *η*-reduce to *x*n.

The following equalities hold:

*η*(*T* )*∗* = *λ→x.zη*(*T*1)*∗ ··· η*(*T*k)*∗ ∪ {⊥}* by def. of *η*(*—*)

= *λ→x.z* nfηe (*E* (*T∗*))’ *···* nfηe (*E* (*T∗*))’ *∪ {⊥}* by co-IH

1 k

= *λ→x.z*nfηe (*E* (*T∗*)) *···* nfηe (*E* (*T∗*))’

1 k

*∪* *{E* (*⊥,* (*λ→x.zT ∗ ··· T∗*) *↓*)*}*’ by def. of *·*’

1 k

= *λ→x.z*nfηe (*E* (*T∗*)) *···* nfηe (*E* (*T∗*)) by def. of *·*’

1 k

*∪ {*nfηe (*E* (*⊥,* (*λ→x.zT ∗ ··· T∗*) *↓*)*}*)’ and of nfη(*—*)

1 k

= nfηe *λ→x.zE* (*T∗*) *···E* (*T∗*))

1

n

*∪ {E* (*⊥,* (*λ→x.zT ∗ ··· T∗*) *↓*)*}* ’ by def. of nfη(*—*)

1

k

= nfηe (*E* (*λ→x.zT ∗ ··· T∗*) *↓*)’ by def. of *E* (*—*)

1 k

= nfηe (*E* (*T∗*))’ by Lemma [5.11](#_bookmark44)*. 2*

**Proposition 5.12** *For all M ∈* Λ*, we have* BTη(*M* )*∗* = nfηe *E* (BT(*M* )*∗*)’*.*

**Proof** Since BTη(*M* )= *η*(BT(*M* )), the result follows directly by Lemma [A.5](#_bookmark83). *2*

**Lemma A.6** *Let M ∈* Λ *be a β-normal form such that M* →η *x. Then for all*

*t ∈f* (*M* )*, we have L*(*t, f* (*M* )) →η*Æ x.*

**Proof** By hypothesis, *M* has the shape *λx*1 *... x*n*.xM*1 *··· M*n (for some *n ≥* 0) such that, for all *i ∈ {*1*,..., n}*, *x /*= *x*i and *M*i is a *β*-normal form such that *M*i →η *x*i. We proceed by induction on *t*. Since *t ∈ f* (*M* ), we have *t* = *λx*1 *... x*n*.xb*1 *··· b*n such that *b*i *∈ M*f(*f* (*M*i)) for every 1 *≤ i ≤ n*. If *n* = 0 we are done. Otherwise, by Definition [5.1](#_bookmark39) we have *L*(*t, f* (*M* )) = *λx*1 *... x*n*.xL*(*b*1*, M*f(*f* (*M*1))) *··· L*(*b*n*, M*f(*f* (*M*n)))*.*

Suppose *b*n = [*t*1*,..., t*k] with *t*j *∈f* (*M*n) for all *j ∈ {*1*,..., k}*.

If *k* =0 then, by Definition [5.1](#_bookmark39), *L*(*b*n*, M*f(*f* (*M*n))) = 1x*n* because *M*n →η *x*n

η(x*n*)

entails that there is *s ∈* S *M*f(*f* (*M*n)) = *f* (*M*n) such that *s* →η*′ x*n. Therefore:

*L*(*t, f* (*M*n)) →η*Æ λx*1 *... x*n*.xL*(*b*1*, M*f(*f* (*M*1))) *··· L*(*b*n*—*1*, M*f(*f* (*M*n*—*1)))1x*n*

η(x*n*)

→η*Æ λx*1 *... x*n*—*1*.xL*(*b*1*, M*f(*f* (*M*1))) *··· L*(*b*n*—*1*, M*f(*f* (*M*n*—*1))) If *k >* 0, then by induction hypothesis *L*(*t*nj*, f* (*M*n)) →η*Æ x*n. Therefore,

*L*(*t, f* (*M*n)) →η*Æ λx*1 *... x*n*.xL*(*b*1*, M*f(*f* (*M*1))) *··· L*(*b*n*—*1*, M*f(*f* (*M*n*—*1)))[*x*k ]

n

→η*Æ λx*1 *... x*n*—*1*.xL*(*b*1*, M*f(*f* (*M*1))) *··· L*(*b*n*—*1*, M*f(*f* (*M*n*—*1))) By iterating this reasoning on *b*1*,..., b*n*—*1 we conclude that *L*(*t, f* (*M* )) →η*Æ x*. *2*

**Lemma 5.13** *Let T* = *λ→xy.zT*1 *··· T*k+1 *be a B¨ohm tree such that T*k+1 *is ﬁnite,*

*T*k+1 →η *y and y ∈/* fv(*zT*1 *··· T*k)*. Then L*(*f* (*T* )) *⇒*η*Æ L*(*f* (*λ→x.zT*1 *··· T*k))*.*

**Proof** We first take *t ∈ f* (*T* ), that is *t* = *λ→xy.zb*1 *··· b*k+1 with *b*i *∈ M*f(*f* (*T*i)), and show that *L*(*t, f* (*T* )) →η*Æ L*(*tj, f* (*λ→x.zT*1 *··· T*k)) holds for *tj* = *λ→x.zb*1 *··· b*k *∈ L*(*f* (*λ→x.zT*1 *··· T*k)). By definition of the labeling *L*(*—*), we have *L*(*t, f* (*T* )) = *λ→xy.zL*(*b*1*, M*f(*f* (*T*1))) *··· L*(*b*k+1*, M*f(*f* (*T*k+1))). By Remark [5.4](#_bookmark40) we have that *y ∈/* fv(*zT*1 *··· T*k) implies *y ∈/* fv(*zL*(*b*1*, M*f(*f* (*T*1))) *··· L*(*b*k*, M*f(*f* (*T*k))).

˜

Suppose that *b*k+1 = [*t*1*,..., t*n], we split into cases depending on *n*.

**Case** *n* = 0. As *T*k+1 →η *y*, then *T*k+1 is *⊥*-free finite tree, and therefore there exists an *s ∈ f* (*T*k+1) without empty bags such that *s* →η*′ y*. Hence

*L*(*b*k+1*, M*f(*f* (*T*k+1))) = *L*(1*, M*f(*f* (*T*k+1))) = 1y since *s ∈* S *M*f(*f* (*T*k+1)) =

η(y)

*f* (*T*k+1). Therefore we have:

*L*(*t, f* (*T* )) = *λ→xy.zL*(*b*1*, M*f(*f* (*T*1))) *··· L*(*b*k+1*, M*f(*f* (*T*k+1)))

= *λ→xy.zL*(*b*1*, M*f(*f* (*T*1))) *··· L*(*b*k*, M*f(*f* (*T*k)))1y

η(y)

*→*η*Æ λ→xy.zL*(*b*1*, M*f(*f* (*T*1))) *··· L*(*b*k*, M*f(*f* (*T*k)))

= *L*(*λ→xy.zb*1 *··· b*k*, f* (*λ→xy.zT*1 *··· T*k))*.*

**Case** *n >* 0**.** Then *t*i *∈ f* (*T*k+1) for 1 *≤ i ≤ n*, and *L*(*b*k+1*, M*f(*f* (*T*k+1))) = [*L*(*t*1*, f* (*T*k+1))*,..., L*(*t*n*, f* (*T*k+1))]. Since *T*k+1 →η *y*, then *T*k+1 is a *⊥*-free finite tree (that is a *β*-normal *λ*-term), so by Lemma [A.6](#_bookmark84) we have *L*(*t*i*, f* (*T*k+1)) →η*Æ y* for every 1 *≤ i ≤ n*. Therefore:

*L*(*t, f* (*T* )) = *λ→xy.zL*(*b*1*, M*f(*f* (*T*1))) *··· L*(*b*k+1*, M*f(*f* (*T*k+1)))

→η*Æ λ→xy.zL*(*b*1*, M*f(*f* (*T*1))) *··· L*(*b*k*, M*f(*f* (*T*k)))[*y*n]

*→*η*Æ λ→xy.zL*(*b*1*, M*f(*f* (*T*1))) *··· L*(*b*k*, M*f(*f* (*T*k)))

Second, we take *s ∈f* (*λ→x.zT*1 *··· T*k), i.e. *s* = *λ→x.zb*1 *··· b*k with *b*i *∈ M*f(*f* (*T*i)), and show that *L*(*t, f* (*T* )) →η*Æ L*(*s, f* (*λ→x.zT*1 *··· T*k)) for *t* = *λ→xy.zb*1 *··· b*k1 *∈f* (*T* ).

As *T*k+1 →η *y*, then *T*k+1 is *⊥*-free finite tree, and therefore there exists an *s ∈ f* (*T*k+1) without empty bags such that *s* →η*′ y*. Thus *L*(1*, M*f(*f* (*T*k+1))) = 1η(y)

y

since *s ∈* S *M*f(*f* (*T*k+1)) = *f* (*T*k+1). Hence, we have:

*L*(*t, f* (*T* )) = *λ→xy.zL*(*b*1*, M*f(*f* (*T*1))) *··· L*(*bj , M*f(*f* (*T*k)))*L*(1*, M*f(*f* (*T*k+1)))

k

= *λ→xy.zL*(*b*1*, M*f(*f* (*T*1))) *··· L*(*b*k*, M*f(*f* (*T*k)))1y

η(y)

*→*η*Æ λ→x.zL*(*b*1*, M*f(*f* (*T*1))) *··· L*(*b*k*, M*f(*f* (*T*k)))

= *L*(*s, f* (*λ→x.zT*1 *··· T*k))*.*

This completes the proof. *2*

**Lemma A.7** *For all B¨ohm tree T the following equality holds:*

*f* ( nfηe *E* (*T* )’)= nfη*Æ L*(*f* (*T* ))’

*∗*

**Proof** We proceed by co-induction on *T* .

If *T* = *⊥*, then the equality follows because *f* (*⊥*)= *∅*.

Otherwise, the B¨ohm tree *T* can be written in a unique way as *T* =

*λx*1 *... x*n*y*1 *... y*m*.zT*1 *··· T*k*T j ··· T j*

(for some *n, m, k ≥* 0) such that:

1 m

* *y*i *∈/* fv(*zT*1 *··· T*k), *T j* is finite and *T j* →η *y*i for all *i ∈ {*1*,..., m}*,

i i

* *x*n *∈* fv(*zT*1 *··· T*k) or *T*k is infinite, or *T*k is finite but does not *η*-reduce to *x*n. Therefore, the following equalities hold:

*f* ( nfηe *E* (*T∗*)’)= *f* ( nfηe *E* ((*λx*1 *... x*n*.zT*1 *··· T*k) *↓*)’) by Lemma [5.11](#_bookmark44)

= *f* ( *λ→x.z*nfηe (*E* (*T∗*)) *···* nfηe (*E* (*T∗*))’ *∪ {* *E* (*⊥, λ→x.zT ∗ ··· T∗*)’*}*) by def. of *E* (*—*)

1 k 1 k

= *f* ( *λ→x.z*nfηe (*E* (*T∗*)) *···* nfηe (*E* (*T∗*))’) *∪f* ( *E* (*⊥, λ→x.zT ∗ ··· T∗*)’) by def. of *f* (*—*)

1 k 1 k

= *λ→x.zM*f (*f* ( nfηe (*E* (*T∗*))’)) *··· M*f (*f* ( nfηe (*E* (*T∗*))’)) *∪f* (*⊥*) by def. of *f* (*—*)

1 k

= *λ→x.zM*f(*f* ( nfηe (*E* (*T∗*))’)) *··· M*f(*f* ( nfηe (*E* (*T∗*))’)) since *f* (*⊥*)= *∅*

1 k

= *λ→x.zM*f( nfη*Æ* (*L*(*f* (*T*1)))’) *··· M*f( nfη*Æ* (*L*(*f* (*T*k)))’) by co-IH

= nfη*Æ* (*λ→x.zM*f(*L*(*f* (*T*1))) *··· M*f(*L*(*f* (*T*k))))’ by def. of nfη*Æ*

= nfη*Æ L*(*λ→x.zM*f(*f* (*T*1)) *··· M*f(*f* (*T*k)))’ by def. of *L*(*—*)

= nfη*Æ L*(*f* (*λ→x.zT*1 *··· T*k))’ by def. of *f* (*—*)

= nfη*Æ L*(*f* (*T* ))’ by Lemma [5.13](#_bookmark46).*2*

**Proposition 5.14** *For all M ∈* Λ*, f* ( nfηe *E* (BT(*M* )*∗*)’)= nfη*Æ L*(*f* (BT(*M* )))’*.*

**Proof** It follows directly from Lemma [A.7](#_bookmark85). *2*