Separating Terms through Multi Types

Adrienne Lancelot

Inria & LIX, École Polytechnique, UMR 7161, Palaiseau
Université Paris Cité, CNRS, IRIF, F-75013, Paris
adrienne.lancelot@inria.fr

Abstract

Intersection types, as any adequate model of the λ-calculus, provide an equational theory on terms, that we dub type equivalence. We study the proof technique to syntactically characterize type equivalence. To do so, we explore an easy setting, namely weak head type equivalence, which is the equational theory induced by weak head non idempotent intersection types. We prove a folklore result: weak head type equivalence coincides with Sangiorgi's normal form bisimilarity. The crucial part of this characterization is to show that terms that are not normal form bisimilar are not type equivalent: we do so by separating pairs of un-bisimilar terms via multi types.

The presentation slightly differs from previous works by Breuvart et al. in that we do not need to introduce term approximants, solely relying on modern coinductive program equivalences.

1 Introduction

Intersection types, developed to study the theory of programming languages, are at the intersection between operational and denotational techniques. While only a certain subset of terminating programs are typable in a simple type system, all terminating programs are typable in an *intersection* type system. Building on this characterization (intersection typable programs are exactly terminating programs), intersection type system may give syntactic presentations of denotational models [Ronchi Della Rocca, 2018, Bono and Dezani-Ciancaglini, 2020].

Non idempotent intersection types, here referred to as *multi types*, have been especially successful syntactic presentations of the relational model, both in Call-by-Name (CbN) [Bucciarelli et al., 2007] and Call-by-Value (CbV) [Ehrhard, 2012]. In this work we focus on another variant, namely *Weak Head Call-by-Name* (further restricting head reduction to forbid reduction under lambdas).

Instead of relating a type system with denotational semantics, we will investigate the *operational* aspect of multi types by syntactically characterizing the equational theory induced on terms by multi types. Such a characterization is already known for CbN: Böhm tree equivalence exactly represents the equational theory induced by idempotent intersection types [Ronchi Della Rocca, 1982] and by non idempotent intersection types [Breuvart et al., 2018]. Both developments introduce term approximants to carefully define the syntactic program equivalence that will match the equational theory induced by the type system. Following modern presentations of coinductive program equivalences [Sangiorgi, 1994, Lassen, 1999], we shall avoid introducing approximants.

Program Equivalences & Normal form bisimilarities. Contextual equivalence is hard to use to prove program equivalences but it is very natural: two programs are contextually equivalent if they behave the same in every possible environment. Normal form bisimilarities have been introduced to proficiently provide proofs of contextual equivalence between programs. Contrarily to the extensional procedure of contextual equivalence, normal form bisimilarity is

an intensional equivalence, that compares the structure of the terms' (possibly infinite) normal forms.

The normal form bisimilarity of interest here is Sangiorgi's normal form bisimilarity, denoted $\simeq_{wh}$. It is obtained from the study of bisimulations in the π-calculus and the translation of the λ-calculus into π [Sangiorgi, 1994]. Later, Lassen related this bisimilarity, there called weak head normal form bisimilarity, to Lévy-Longo tree equivalence [1999].

Weak Head Multi Types. Adequate intersection types for weak head call-by-name evaluation were first introduced in an idempotent setting by Dezani-Ciancaglini et al. [2004] and then refined quantitatively by Bucciarelli et al. [2017]. We shall use the latter presentation, via multi types defined later in Section 2. The quantitative aspect will play a crucial role, as explained in Section 3.

Type Equivalence. Given an intersection type system, one can define what we call here *type equivalence* $\simeq_{type}$, which is the equational theory induced by the model syntactically represented by the type system. Two terms are type equivalent if they are typable by exactly the same pairs of typing contexts and types. While it is easier to prove type equivalence than contextual equivalence on certain pairs of terms, it is still quantified universally and, unlike normal form bisimilarities, does not provide a straightforward proof technique. Type equivalence is included in contextual equivalence, but this inclusion is in general strict.

In the case of weak head multi types, type equivalence $\simeq_{type}$ does not validate well-known examples of equations validated by contextual equivalence $\simeq_C$ (that are also examples of incompleteness for Sangiorgi's normal form bisimilarity): $x\lambda y.xy \simeq_C xx$ but $x\lambda y.xy \not\simeq_{type} xx$ (and $x\lambda y.xy \not\simeq_{wh} xx$) [Abramsky and Ong, 1993, Sangiorgi, 1994].

This Work. Our main result is that type equivalence induced by weak head multi types coincides with Sangiorgi's normal form bisimilarity. This characterization is not surprising and probably folklore for experts of the topic. We split the equivalence proof in two directions:

$$\text{NF Bisimilarity} \underset{\Longleftarrow \text{ Type-Böhm Out}}{\overset{\Longrightarrow \text{ Derivation Transfer}}{=\!=\!=}} \text{Type Equivalence}$$

Derivation transfer shows that normal form bisimilar terms are typable by the same pairs of typing contexts and types (in fact, using very much the same derivations). *Type-Böhm out technique* provides an explicit construction of a pair of a typing context and a type separating two terms that are not normal form bisimilar. It is similar to the separation construction of Breuvart et al. [2018] and is reminiscent of the well-known Böhm out technique (where from two terms that are not normal form bisimilar, one builds a context separating them for contextual equivalence) [Barendregt, 1984].

An interesting consequence of this theorem is an alternative proof of the fact that Sangiorgi's normal form bisimilarity is included in contextual equivalence. Such a proof is not usually trivial and requires some work [Lassen, 1999].

Coinduction vs. Approximants. There are various related works on characterizing syntactically the equational theory of a model, we mention some in the next paragraph. To the best of our knowledge, characterizations regarding intersection types all mention term approximants. Approximants are an inductive way to look at infinitary behaviors and were the core of the

original definition of Böhm trees. Coinduction is a more modern way to define infinitary computation and it is particularly successful in defining modern program equivalences like normal form bisimilarity.

Related Work. Our construction is fairly similar to Breuvart et al. [2018] where a similar result is proved for head reduction and the appropriate multi types, but for the fact that we consider weak head reduction. We follow their simplification of the proof by introducing a countable number of ground types, which allows to easily separate terms.

For idempotent intersection types, a syntactical characterization of the equational theory behind CbN types was made by Ronchi Della Rocca [1982] using principal types–the syntactical program equivalence is the same as Breuvart et al. [2018].

Recent work on CbN idempotent intersection types by Polonsky and Statman [2022] discuss uniqueness typings to separate terms up to $\beta\eta$ equivalence. It would be interesting to see if their results generalize to normal form bisimilarity.

Future Work. This work was mostly done to get familiar with the proof technique. The next step is to work it out in call-by-value to be able to syntactically characterize Ehrhard's call-by-value relational semantics [2012]. Preliminary results containing only derivation transfer are already available on arXiv [Accattoli et al., 2023].

Relating all these program equivalences to contextual equivalence is also of interest. As we have mentioned above, both weak head type equivalence and Sangiorgi's normal form bisimilarity distinguish strictly more than contextual equivalence. Boudol relates normal form bisimilarity and contextual equivalence restricted to the image of the CPS translation [2000]. We would like to see whether it is possible to directly relate weak head type equivalence and that restriction of contextual equivalence.

2 Weak Head Reduction & Multi Types

Weak Head Reduction We focus on a specific reduction of the lambda calculus, weak head reduction, a variant of head reduction that does not reduce under lambdas.

$$\textsc{Terms} \quad \Lambda \ni t, u, s \quad ::= \quad x \mid \lambda x.t \mid tu \qquad \textsc{Weak Head Ctxs} \quad E \quad ::= \quad \langle\cdot\rangle \mid Et$$

Weak head reduction, denoted $\to_{\mathsf{wh}}$, is the closure of the beta rule $(\lambda x.t)u \mapsto_\beta t\{x \leftarrow u\}$ under applicative *weak head* contexts. In essence, all $\to_{\mathsf{wh}}$ reduction are of the shape $(\lambda x.t)u\ s_1 \dots s_k \to_{\mathsf{wh}} t\{x \leftarrow u\}\ s_1 \dots s_k$ for $k \geq 0$.

Normal forms are exactly characterized by the following grammar, separating normal forms into rigid terms and abstractions:

$$\textsc{Rigid Terms} \quad r, r' ::= x \mid rt \qquad\qquad \textsc{Weak Head Normal Forms} \quad w, w' ::= \lambda x.t \mid r$$

Weak Head Multi Types. We follow the definitions of multi types for weak head reduction as designed by Bucciarelli et al. [2017], recalled in Fig. 1.

Multi Types. Multi types M are finite multisets of linear types L that are either base types $\star_k$ or arrow types $M \multimap L$. In the original presentation of Bucciarelli et al. [2017], there are distinct ground types and an abstraction types $\sharp$: we unify both for simplicity. Note that there is a countable number of distinct base types, following Breuvart et al. [2018] to simplify separating terms by types.

$$\begin{array}{rrcl} \textsc{Linear Types} & L, L' & ::= & \star_k \mid M \multimap L \quad k \in \mathbb{N} \\ \textsc{Multi Types} & M, N & ::= & [L_1, \dots, L_n] \quad n \geq 0 \end{array}$$

$$\frac{}{x : [L] \vdash x : L}\ \mathsf{ax} \qquad \frac{}{\emptyset \vdash \lambda x.t : \star_k}\ \lambda_k \qquad \frac{\Gamma, x : M \vdash t : L}{\Gamma \vdash \lambda x.t : M \multimap L}\ \lambda$$

$$\frac{\Gamma \vdash t : [L_i]_{i \in I} \multimap L' \quad (\Gamma_i \vdash u : L_i)_{i \in I} \quad I \text{ finite}}{\Gamma \uplus \Delta \vdash tu : L'}\ @$$

Figure 1: $\mathcal{WH}$ multi types.

Results about Multi Types. Intersection types are mostly known for the fact that typability exactly captures normalizable terms. This system is no exception, as a term is typable in $\mathcal{WH}$ if and only if it is *weak head* normalizable.

Theorem 2.1 (Characterization of Termination)**.** *A term t is typable in $\mathcal{WH}$ iff $t \to_{\mathsf{wh}}^* n$ where n is a weak head normal form.*

Historically such a proof for idempotent intersection types is obtained by reducibility arguments but one can avoid this proof technique if considering multi types. The main selling point of multi types, the *non idempotent* variant of intersection types, is to obtain *quantitative* results from typing judgments. The most famous example is the following statement of *quantitative subject reduction*, that allows to show that any type of a term is stable by reduction, and the derivation of the reduct will be of a strictly smaller size. The size of a derivation $|\pi|$ is its total number of rules.

Proposition 2.2 (Quantitative Subject Reduction, Bucciarelli et al. [2017])**.** *If $\pi \triangleright \Gamma \vdash t : L$ and $t \to_{\mathsf{wh}} t'$ then $\pi' \triangleright \Gamma \vdash t' : L$ with $|\pi'| < |\pi|$.*

From quantitative subject reduction, it is typical to deduce (without reducibility arguments!) that the intersection type system is correct, *i.e.* all typable terms are normalizing. For the converse implication to hold, thus completing the characterization of termination, one can easily deduce it from (quantitative) subject expansion but the quantitative argument plays no role in that proof. Quantitative arguments are also sometimes taken a step further, introducing alternative type systems that exactly measure the length of the reduction sequence to normal forms [Accattoli et al., 2018].

The Equational Theory Induced by Multi Types. In this work, we focus on the program equivalence induced by multi types. Indeed, multi types induce a model, that we shall not discuss here, and every model induces an equivalence relation on terms (by relating terms with the same interpretation in the model). We introduce the *type preorder*, that exactly rephrases the (in)equational theory induced by weak head multi types.

Definition 2.3 (Type Preorder)**.** *We define a preorder on terms based on their type derivations:*

- *$t \precsim_{type} t'$ if for all Γ, M such that there exists $\pi : \Gamma \vdash t : M$ then there exists $\pi' : \Gamma \vdash t' : M$.*
- *Type equivalence is defined by symmetry: $t \simeq_{\mathcal{WH}} t'$ iff $t \precsim_{type} t'$ and $t' \precsim_{type} t$.*

It is easy to check that $\precsim_{type}$ is indeed a preorder (reflexive and transitive). Furthermore by subject reduction and subject expansion it is invariant by $\to_{\mathsf{wh}}$ and *adequate*. As it is also stable by contexts, the type preorder satisfies the usual definition of an (in)equational theory [Barendregt, 1984, Barendregt and Manzonetto, 2022].

Proposition 2.4. *The type preorder enjoys the following properties:*

1. Adequacy: *if* $t \precsim_{type} t'$ *and* t *is* $\to_{\mathsf{wh}}$*-normalizing then* t' *is* $\to_{\mathsf{wh}}$*-normalizing.*
2. Monotonicity: *if* $t \precsim_{type} t'$ *then* $\forall C, C\langle t\rangle \precsim_{type} C\langle t'\rangle$.

As a direct corollary, the type preorder is included in the contextual preorder, of which we omit the definition here.

3 A syntactic characterization of type equivalence

In this section, we recall the definition of Sangiorgi's normal form (bi)similarity and show the correspondence with the type preorder.

First, from a relation $\mathcal{R}$ on terms we define its closure on rigid terms as $\langle\mathcal{R}\rangle_{nf}$. Formally, $\langle\mathcal{R}\rangle_{nf}$ is defined, by induction, as the smallest relation including the three following rules:

$$\frac{}{x\ \langle\mathcal{R}\rangle_{nf}\ x} \qquad \frac{r\ \langle\mathcal{R}\rangle_{nf}\ r' \quad t\ \mathcal{R}\ u}{rt\ \langle\mathcal{R}\rangle_{nf}\ r'u} \qquad \frac{t\ \mathcal{R}\ u}{\lambda x.t\ \langle\mathcal{R}\rangle_{nf}\ \lambda x.u}$$

Definition 3.1 (Weak head normal form simulation)**.** *A relation on terms* $\mathcal{R}$ *is a weak head normal form (whnf) simulation if for all* t, t' *such that* $t\ \mathcal{R}\ t'$ *either* t *does not* $\to_{\mathsf{wh}}$*-terminate, or* $t \to_{\mathsf{wh}}$*-reduces to a normal form* w *and* $t' \to_{\mathsf{wh}}$*-reduces to a normal form* w' *such that* $w\ \langle\mathcal{R}\rangle_{nf}\ w'$.

Whnf similarity, noted $\precsim_{wh}$*, is defined, by coinduction, as the largest whnf simulation.*

The definition here is slightly different than Sangiorgi's and Lassen's well-known presentations [1994, 1999] as we make explicit an inductive definition for rigid terms, via $\langle\mathcal{R}\rangle_{nf}$.

Type equivalence exactly matches normal form bisimilarity. Now, we characterize exactly the type preorder by whnf similarity, *i.e.* $\precsim_{wh}=\precsim_{type}$. We call *correctness* $\precsim_{wh}\subseteq\precsim_{type}$, the easy part, and *completeness* $\precsim_{wh}\supseteq\precsim_{type}$, the difficult part.

Correctness via Derivation Transfer. We shall first show correctness and we do so via the following *derivation transfer* proposition. It says that if two terms are whnf similar, then any type derivation of t immediately transfers to t'. We show such a result via induction on the size of the derivation. A crucial ingredient is that subject reduction does not increase the size of the derivation, otherwise the induction argument would not work. The quantitative aspect of the subject reduction proof is therefore crucial and justifies the need of *multi* types.

Proposition 3.2 (Derivation Transfer)**.** *Let* $\mathcal{R}$ *a whnf simulation. If* $t\ \mathcal{R}\ t'$ *and there exists a derivation* $\pi : \Gamma \vdash t : M$ *then there exists a derivation* $\pi' : \Gamma \vdash t' : M$.

We deduce correctness, *i.e.* $\precsim_{wh}\subseteq\precsim_{type}$, easily from derivation transfer.

Completeness via Type-Böhm Out. To prove completeness, *i.e.* that $\precsim_{type}\subseteq\precsim_{wh}$, we will use coinduction. We first show that $\precsim_{type}$ is a whnf simulation, which implies it is included in the largest whnf simulation, namely whnf similarity $\precsim_{wh}$.

Separating terms with types. We prove by contrapositive that the type preorder is a whnf simulation: if two terms are type different after one step of simulation then there exists a typing context and a type that separates them before the simulation step. Broadly, if $t \not\precsim_{wh} u$ then there exists Γ, L such that $\Gamma \vdash t : L$ but $\Gamma \nvdash u : L$. We can explicitly show how to build such a pair (Γ, L), not featured here for lack of space, and we shall present this construction on examples during the talk.

References

S. Abramsky and C. H. L. Ong. Full Abstraction in the Lazy Lambda Calculus. *Information and Computation*, 105(2):159–267, 1993. ISSN 0890-5401. doi: https://doi.org/10.1006/inco.1993.1044. URL https://www.sciencedirect.com/science/article/pii/S0890540183710448.

B. Accattoli, S. Graham-Lengrand, and D. Kesner. Tight typings and split bounds. *PACMPL*, 2(ICFP):94:1–94:30, 2018. doi: 10.1145/3236789.

B. Accattoli, A. Lancelot, and C. Faggian. Normal form bisimulations by value, 2023.

H. Barendregt and G. Manzonetto. *A Lambda Calculus Satellite*. College Publications, 2022. ISBN 978-1-84890-415-6. URL https://www.collegepublications.co.uk/logic/mlf/?00035.

H. P. Barendregt. *The Lambda Calculus – Its Syntax and Semantics*, volume 103 of *Studies in logic and the foundations of mathematics*. North-Holland, 1984. ISBN 9780444867483.

V. Bono and M. Dezani-Ciancaglini. A tale of intersection types. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '20, page 7–20, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450371049. doi: 10.1145/3373718.3394733. URL https://doi.org/10.1145/3373718.3394733.

G. Boudol. On the semantics of the call-by-name cps transform. *Theoretical Computer Science*, 234(1):309–321, 2000. ISSN 0304-3975. doi: https://doi.org/10.1016/S0304-3975(99)00302-3. URL https://www.sciencedirect.com/science/article/pii/S0304397599003023.

F. Breuvart, G. Manzonetto, and D. Ruoppolo. Relational Graph Models at Work. *Logical Methods in Computer Science*, Volume 14, Issue 3, July 2018. doi: 10.23638/LMCS-14(3:2)2018. URL https://lmcs.episciences.org/4701.

A. Bucciarelli, T. Ehrhard, and G. Manzonetto. Not enough points is enough. In J. Duparc and T. A. Henzinger, editors, *Computer Science Logic*, pages 298–312, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-74915-8.

A. Bucciarelli, D. Kesner, and D. Ventura. Non-idempotent intersection types for the lambda-calculus. *Logic Journal of the IGPL*, 25(4):431–464, 2017. doi: 10.1093/jigpal/jzx018.

M. Dezani-Ciancaglini, S. Ghilezan, and S. Likavec. Behavioural inverse limit λ-models. *Theoretical Computer Science*, 316(1):49–74, 2004. ISSN 0304-3975. doi: https://doi.org/10.1016/j.tcs.2004.01.023. URL https://www.sciencedirect.com/science/article/pii/S0304397504000763. Recent Developments in Domain Theory: A collection of papers in honour of Dana S. Scott.

T. Ehrhard. Collapsing non-idempotent intersection types. In P. Cégielski and A. Durand, editors, *Computer Science Logic (CSL'12) - 26th International Workshop/21st Annual Conference of the EACSL, CSL 2012, September 3-6, 2012, Fontainebleau, France*, volume 16 of *LIPIcs*, pages 259–273. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012. doi: 10.4230/LIPIcs.CSL.2012.259. URL https://doi.org/10.4230/LIPIcs.CSL.2012.259.

S. B. Lassen. Bisimulation in untyped lambda calculus: Böhm trees and bisimulation up to context. 20:346–374, 1999. doi: 10.1016/S1571-0661(04)80083-5. URL https://doi.org/10.1016/S1571-0661(04)80083-5.

A. Polonsky and R. Statman. On sets of terms having a given intersection type. *Logical Methods in Computer Science*, Volume 18, Issue 3, Sept. 2022. ISSN 1860-5974. doi: 10.46298/lmcs-18(3:35)2022. URL http://dx.doi.org/10.46298/lmcs-18(3:35)2022.

S. Ronchi Della Rocca. Characterization theorems for a filter lambda model. *Information and Control*, 54(3):201–216, 1982. ISSN 0019-9958. doi: https://doi.org/10.1016/S0019-9958(82)80022-3. URL https://www.sciencedirect.com/science/article/pii/S0019995882800223.

S. Ronchi Della Rocca. Intersection Types and Denotational Semantics: An Extended Abstract. In S. Ghilezan, H. Geuvers, and J. Ivetic, editors, *22nd International Conference on Types for Proofs and Programs (TYPES 2016)*, volume 97 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 2:1–2:7, Dagstuhl, Germany, 2018. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-065-1. doi: 10.4230/LIPIcs.TYPES.2016.2. URL https://drops-dev.dagstuhl.de/entities/document/10.4230/LIPIcs.TYPES.2016.2.

D. Sangiorgi. The lazy lambda calculus in a concurrency scenario. *Information and Computation*, 111(1):120–153, 1994. ISSN 0890-5401. doi: https://doi.org/10.1006/inco.1994.1042. URL https://www.sciencedirect.com/science/article/pii/S089054018471042X.