From Normalization to Typability via Subject Expansion

Andrej Dudenhefner[1]

TU Dortmund University, Germany
andrej.dudenhefner@cs.tu-dortmund.de

Abstract

We use a recently developed notion of IK'-reduction in order to show that each strongly normalizing term is typable in the Coppo-Dezani intersection type assignment system.

1 Introduction

In the early 1980s Coppo and Dezani-Ciancaglini presented an intersection type assignment system (CD) [3] as an extension of the simple type system. In their seminal work, CD is shown to assign types to strongly normalizing terms in the λ-**I**-calculus. Afterwards, Pottinger [7] has shown that any strongly normalizing term is typable in CD. As keenly remarked by Barendregt, Dekkers, and Statman [2, Remark 17.2.16 (ii)], there are many proofs of this fact in literature, most of which (including Pottinger's) are incorrect.

The present work gives an alternative approach to show that strongly normalizing terms are typable in CD. The key idea is to use a recently developed notion of IK'-reduction [4], which allows for subject expansion properties. Any strongly normalizing term M reduces to a typable normal form via the IK'-reduction. Therefore, subject expansion properties imply typability of M. In order to eliminate any doubt, the proof is mechanized in constructive logic using the Coq proof assistant [8].

2 Coppo-Dezani Intersection Type Assignment System

The Coppo-Dezani type assignment system [3, Section 2] is an extension of the simple type system, having finite, non-empty sets of types as premises. In this section, let us briefly recall the basic definitions.

We denote λ-*terms* (Definition 1) by M, N, where *term variables* are denoted by x, y.

Definition 1.
(λ-Terms) $M, N \quad ::= \quad x \mid \lambda x.M \mid M\,N$

We denote *intersection types* (Definition 2) by A, B, finite, non-empty sets of intersection types by σ, τ, and *type atoms* by a, b. The empty intersection (usually called the universal type ω) is not part of the definition, which otherwise would hinder a characterization of strongly normalizing terms via typability.

Definition 2.
(Intersection Types) $A, B \quad ::= \quad a \mid \sigma \to A$
(Sets) $\sigma, \tau \quad ::= \quad \{A_1, \ldots, A_n\}$ *where* $n \geq 1$

An *environment*, denoted by Γ, is a finite set of *type assumptions* having the shape $x : \sigma$ for distinct term variables.

Definition 3.
(Environments) $\Gamma \quad ::= \quad \{x_1 : \sigma_1, \ldots, x_n : \sigma_n\}$ *where* $x_i \neq x_j$ *for* $i \neq j$
For $\Gamma = \{x_1 : \sigma_1, \ldots, x_n : \sigma_n\}$ *we have*
(Extension) $\Gamma, x : \sigma \quad := \quad \Gamma \cup \{x : \sigma\}$ *where* $x \notin \{x_1, \ldots, x_n\}$

The rules of the Coppo-Dezani type assignment system for *judgements* of shape $\Gamma \vdash M : A$ are given below.

Definition 4 (Coppo-Dezani Type Assignment System)**.**

$$\frac{A \in \sigma}{\Gamma, x : \sigma \vdash x : A}\ (\text{Var}) \qquad \frac{\Gamma, x : \sigma \vdash M : A}{\Gamma \vdash \lambda x.M : \sigma \to A}\ (\to\text{I})$$

$$\frac{\Gamma \vdash M : \{A_1, \ldots A_n\} \to A \qquad \Gamma \vdash N : A_1 \quad \ldots \quad \Gamma \vdash N : A_n}{\Gamma \vdash M\ N : A}\ (\to\text{E})$$

We say that a term M is *typable* if $\Gamma \vdash M : A$ for some environment Γ and type A.

3 IK'-Reduction

In this section, we recall common notions of reductions and give the novel K'-reduction [4]. A substitution in the term M of the free variable x by the term N is denoted $M[x := N]$. The set of free variables in a term M is denoted $\text{FV}(M)$.

Definition 5.
(β-Reduction) *contextual closure of* $(\lambda x.M)\, N \to_\beta M[x := N]$
(I-Reduction) *contextual closure of* $(\lambda x.M)\, N \to_I M[x := N]$ *if* $x \in \text{FV}(M)$
(K-Reduction) *contextual closure of* $(\lambda x.M)\, N \to_K M[x := N]$ *if* $x \notin \text{FV}(M)$

A term is in *β-normal form*, if it cannot be β-reduced.

Lemma 6 ([3, Theorem 3])**.** *If M is in β-normal form, then M is typable.*

Definition 7 (K'-Reduction)**.**

1. *If N is in β-normal form and $x \notin \text{FV}(M)$, then $(\lambda x.M)\, N \to_{K'} M$.*
2. *If $M \to_{K'} N$, then $\lambda x.M \to_{K'} \lambda x.N$.*
3. *If $N_1 \to_{K'} N_2$, then $x\, M_1 \ldots M_n\, N_1 \to_{K'} x\, M_1 \ldots M_n\, N_2$ where $n \geq 0$.*
4. *If $N_1 \to_{K'} N_2$ and $x \notin \text{FV}(M)$, then $(\lambda x.M)\, N_1 \to_{K'} (\lambda x.M)\, N_2$.*
5. *If $M_1\, M_2 \to_{K'} M_3$, then $M_1\, M_2\, N \to_{K'} M_3\, N$.*

For the union of $\to_I$ and $\to_{K'}$, denoted $\to_{IK'}$, we have the following properties.

Lemma 8 ([4, Lemma 5])**.** *If $M \to_\beta N$, then there exists N' such that $M \to_{IK'} N$.*

A term M is *strongly β-normalizing*, if any chain of consecutive β-reduction steps starting from M is finite.

Corollary 9. *If M is strongly β-normalizing, then there exists N in β-normal form such that $M \to_{IK'}^* N$.*

4 Subject Expansion

This section contains subject expansion properties for the I-reduction and the K'-reduction.

Lemma 10 (Subject Expansion for $\to_I$ [3, Theorem 4])**.** *If $M \to_I N$ and $\Gamma \vdash N : A$, then $\Gamma \vdash M : A$.*

Without the universal type ω at our disposal, we do not have subject expansion for $\to_K$, illustrated in the following Example 11.

Example 11. *Let $M := (\lambda x.\lambda y.y)\,((\lambda z.z\,z)\,(\lambda z.z\,z))$ and $N := \lambda y.y$. We have $M \to_K N$ and $\emptyset \vdash N : \{a\} \to a$. However, the term M is not typable.*

Conveniently, K'-reduction allows for (carefully stated) subject expansion.

Lemma 12 (Subject Expansion wrt. $\to_{K'}$)**.** *If $M \to_{K'} N$ and $\Gamma \vdash N : A$, then*

- *if M is not an abstraction, then $\Gamma' \vdash M : A$ for some environment Γ';*
- *if M is an abstraction, $\Gamma' \vdash M : A'$ for some environment Γ' and some type A'.*

Proof Sketch. By induction on the inductive definition of the K'-reduction.

1. $(\lambda x.M')\,N' \to_{K'} M'$ such that N' is in β-normal form and $x \notin \mathrm{FV}(M')$:
   Easy because N' is in β-normal form, and therefore typable.
2. $\lambda x.M' \to_{K'} \lambda x.N'$ such that $M' \to_{K'} N'$:
   Follows from the induction hypothesis.
3. $x\,M_1 \dots M_n\,N_1 \to_{K'} x\,M_1 \dots M_n\,N_2$ such that $N_1 \to_{K'} N_2$:
   Follows from the induction hypothesis by weakening the type assigned to the term variable x in the environment.
4. $(\lambda x.M')\,N_1 \to_{K'} (\lambda x.M')\,N_2$ such that $N_1 \to_{K'} N_2$ and $x \notin \mathrm{FV}(M')$:
   Follows from the induction hypothesis by changing the type of the abstracted term variable x where $x \notin \mathrm{FV}(M')$.
5. $M_1\,M_2\,N \to_{K'} M_3\,N$ such that $M_1\,M_2 \to_{K'} M_3$:
   Follows from the induction hypothesis noting that $M_1\,M_2$ is not an abstraction. ☐

Since IK'-reduction cannot get stuck on β-reducible terms and terms in β-normal form are typable, we obtain from the above subject expansion properties that any strongly β-normalizing term is typable.

Theorem 13. *If M is strongly β-normalizing, then M is typable.*

Proof. By Corollary 9 a strongly β-normalizing term M reduces to a β-normal form N via IK'-reduction. By Lemma 6 the term N is typable. By induction on the number of IK'-reduction steps together with Lemma 10 and Lemma 12 we obtain typability of M. ☐

Since typable terms are strongly β-normalizing [6, Theorem 3.7], any IK'-reduction strategy is perpetual [9], i.e. preserves infinite reduction paths.

Remark 14 ([4, Theorem 37])**.** *If $M \to_{IK'} N$ and N is in β-normal form, then the term M is strongly β-normalizing.*

Corollary 15. *The reduction strategy F_∞ [1, Definition 13.4.1] is perpetual.*

The above results regarding the IK'-reduction are mechanized using the Coq proof assistant. The mechanization is used in the Coq library of undecidability proofs [5] to establish a many-one reduction from strong β-normalization to typability in the Coppo-Dezani type assignment system[1]. The particular statements[2] are as follows.

Mechanized Lemma 10 where `stepI` mechanizes $\to_I$:

```
Lemma stepI_expansion M N Gamma t :
  stepI M N -> type_assignment Gamma N t -> type_assignment Gamma M t.
```

Mechanized Lemma 12 where `stepK` mechanizes $\to_{K'}$:

```
Lemma stepK_expansion M N Gamma t :
  stepK M N -> type_assignment Gamma N t ->
    match M with
    | var _ => exists Gamma', type_assignment Gamma' M t
    | app _ _ => exists Gamma', type_assignment Gamma' M t
    | lam _ => typable M
    end.
```

Mechanized Theorem 13 where `sn` mechanizes strong normalization:

```
Lemma sn_type_assignment M :
  sn M -> exists Gamma t, type_assignment Gamma M t.
```

Mechanized Remark 14 where `steps'` mechanizes $\to^*_{IK'}$:

```
Theorem wn_step'_sn_step (M N : term) :
  steps' M N -> normal_form N -> sn M.
```

References

[1] Hendrik Pieter Barendregt. *The lambda calculus - its syntax and semantics*, volume 103 of *Studies in logic and the foundations of mathematics*. North-Holland, 1985.

[2] Hendrik Pieter Barendregt, Wil Dekkers, and Richard Statman. *Lambda Calculus with Types*. Perspectives in logic. Cambridge University Press, 2013.

[3] Mario Coppo and Mariangiola Dezani-Ciancaglini. An extension of the basic functionality theory for the λ-calculus. *Notre Dame Journal of Formal Logic*, 21(4):685–693, 1980.

[4] Andrej Dudenhefner and Daniele Pautasso. Mechanized subject expansion in uniform intersection types for perpetual reductions. In *FSCD 2024*. (in publication), 2024.

[5] Yannick Forster, Dominique Larchey-Wendling, Andrej Dudenhefner, Edith Heiter, Dominik Kirst, Fabian Kunze, Gert Smolka, Simon Spies, Dominik Wehr, and Maximilian Wuttke. A Coq Library of Undecidable Problems. In *The Sixth International Workshop on Coq for Programming Languages (CoqPL 2020)*, 2020.

[6] Silvia Ghilezan. Strong normalization and typability with intersection types. *Notre Dame J. Formal Log.*, 37(1):44–52, 1996.

[7] Garrel Pottinger. A type assignment for the strongly normalizable λ-terms. *To HB Curry: essays on combinatory logic, lambda calculus and formalism*, pages 561–577, 1980.

[8] The Coq Development Team. The Coq proof assistant, July 2023.

[9] Femke van Raamsdonk, Paula Severi, Morten Heine Sørensen, and Hongwei Xi. Perpetual reductions in lambda-calculus. *Inf. Comput.*, 149(2):173–225, 1999.

[1] https://github.com/uds-psl/coq-library-undecidability/blob/coq-8.19/theories/IntersectionTypes/CD.v

[2] https://github.com/uds-psl/coq-library-undecidability/blob/coq-8.19/theories/IntersectionTypes/Reductions/SNclosed_to_CD_TYP.v