PUBLICATIONS DE L'INSTITUT MATHÉMATIQUE
Nouvelle série, tome 82(96) (2007), 85–91 DOI 102298/PIM0796085G

INTERSECTION TYPES FOR λ^{Gtz}-CALCULUS

Silvia Ghilezan and Jelena Ivetić

Abstract. We introduce an intersection type assignment system for Espirito-Santo's λ^{Gtz}-calculus, a term calculus embodying the Curry–Howard correspondence for the intuitionistic sequent calculus. We investigate basic properties of this intersection type system. Our main result is Subject reduction property.

1. Introduction

Recently, several extensions of λ-calculus were developed in order to extend the Curry-Howard correspondence to intuitionistic sequent calculus. Herbelin [**9**] proposed the first "sequent" λ-calculus, named $\bar{\lambda}$, for which bijective correspondence between normal simply typed terms and cut-free proofs of the appropriate restriction of the Gentzen's LJ was obtained. However, this bijection failed to extend to sequent calculus with cuts. After that, intuitionistic sequent λ-calculi were proposed by Barendregt and Ghilezan [**1**], Dyckhoff and Pinto [**4**], Espirito-Santo and Pinto [**5**], among others. One of the most recently proposed systems is λ^{Gtz}-calculus, developed by Espirito-Santo [**6**], whose simply typed version corresponds to the sequent calculus for intuitionistic implicational logic.

Intersection type assignment systems were introduced by Coppo and Dezani [**2**]. These systems characterize exactly the strongly normalizing λ-terms (proved in Pottinger [**13**], Ghilezan [**7**], Krivine [**10**]) and that property couldn't be obtained with basic simply typed λ-calculus. Since then, intersection types were introduced in several extensions of λ-calculus, like in Lengrand's et al. [**11**] calculus with explicit substitutions, Matthes's [**12**] calculus with generalized applications and Dougherty's et al. [**3**] calculus for classical logic, each time in order to characterize strong normalization.

In this paper, we introduce intersection type assignment to slightly modified Espirito-Santo's intuitionistic sequent λ^{Gtz}-calculus. The paper is organized as follows. In Section 2 untyped λ^{Gtz}-calculus is introduced. In Section 3 we propose a new, intersection type assignment system named $\lambda^{\mathsf{Gtz}}\cap$ and investigate some basic properties of this system. In Section 4 Subject reduction property is proved and

2000 *Mathematics Subject Classification*: Primary 03B40; Secondary 68N18.

Section 5 proposes some future work toward characterization of strong normalization in this calculus, which is our main goal.

2. Syntax of λ^{Gtz}

The abstract syntax of λ^{Gtz} is given by:

$$\begin{array}{ll}\text{(Terms)} & t,u,v ::= x \mid \lambda x.t \mid tk \\ \text{(Contexts)} & k ::= [x]t \mid u :: k\end{array}$$

We distinguish *terms*, which could be variables, abstraction or application (also called *cut*) and *contexts*, which are either a *selection* or linear left introduction (usually called *cons*). The intuition is that a context is actually a list of arguments, since cut is of the form tk, i.e., a function applied to list. A list is made out of a term by the selection operator, and new elements are added by cons. An empty list is of the form $[x]x$ (abbreviated with $[\,]$).

In $\lambda x.t$ and $[x]t$, t is the scope of the binders λx and $[x]$, respectively. In order to save parentheses, we let the scope of binders extend to the right as much as possible.

Definition 2.1. The set of free variables of a term or a context, Fv( ), is defined with:

$$\begin{aligned}\mathrm{Fv}(x) &= \{x\}\\ \mathrm{Fv}(\lambda x.t) &= \mathrm{Fv}(t) \setminus \{x\}\\ \mathrm{Fv}(tk) &= \mathrm{Fv}(t) \cup \mathrm{Fv}(k)\\ \mathrm{Fv}([x]t) &= \mathrm{Fv}(t) \setminus \{x\}\\ \mathrm{Fv}(t :: k) &= \mathrm{Fv}(t) \cup \mathrm{Fv}(k)\end{aligned}$$

Remark 2.1. We can see that free variables in λ^{Gtz} calculus are those that are not bound neither by abstraction nor by selection operator and Barendregt's convention should be applied in both cases.

Reduction rules of λ^{Gtz} are as follows:

$$\begin{array}{lr}(\beta) & (\lambda x.t)(u :: k) \to t\langle x := u\rangle k\\ (\pi) & (tk)k' \to t(k@k')\\ (\sigma) & t[x]v \to v\langle x := t\rangle\\ (\mu) & [x]xk \to k, \text{ if } x \notin k\end{array}$$

where

- $x\langle x := u\rangle = u$; $y\langle x := u\rangle = y$;
- $(\lambda y.t)\langle x := u\rangle = \lambda y.t\langle x := u\rangle$;
- $(tk)\langle x := u\rangle = t\langle x := u\rangle k\langle x := u\rangle$;
- $([y]v)\langle x := u\rangle = [y]v\langle x := u\rangle$;
- $(v :: k)\langle x := u\rangle = v\langle x := u\rangle :: k\langle x := u\rangle$;
- $(u :: k)@k' = u :: (k@k')$;
- $([x]v)@k' = [x]vk'$.

Normal forms of λ^{Gtz} are:

$$\begin{array}{ll}\text{(Terms)} & t_{nf}, u_{nf}, v_{nf} = x_{nf} \mid \lambda x.t_{nf} \mid x(u_{nf} :: k_{nf}) \\ \text{(Contexts)} & k_{nf} = [x]t_{nf} \mid t_{nf} :: k_{nf}\end{array}$$

If application is seen as cut, then reductions aim at eliminating cuts, i.e., only trivial cuts are allowed in normal forms.

3. Intersection types for λ^{Gtz}

Definition 3.1. The set of types Types, ranged over by $A, B, C, \dots, A_1, \dots$, is inductively defined by: $A, B ::= p \mid A \to B \mid A \cap B$, where p ranges over a denumerable set of type atoms.

Definition 3.2. (i) A basic type assignment is an expression of the form $x : A$, where x is a term variable and A is a type.

(ii) A basis Γ is a set of basic type assignments, where all term variables are different.

(iii) There are two kinds of type assignment: $\Gamma \vdash t : A$ for typing terms; $\Gamma; B \vdash k : A$ for typing contexts.

The special place between the symbols ; and $\vdash$ is called the *stoup*. It was proposed by Girard [**8**]. Stoup contains a selected formula, the one with which we continue computation.

The following typing system for λ^{Gtz} is named $\lambda^{\mathsf{Gtz}}\cap$.

$$\frac{}{\Gamma, x : \bigcap A_i \vdash x : A_i \quad i \geqslant 1}\ (\mathrm{Ax})$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B}\ (\to_R) \qquad \frac{\Gamma \vdash u : A_i \quad i = 1, \dots, n \quad \Gamma; B \vdash k : C}{\Gamma; \bigcap A_i \to B \vdash u :: k : C}\ (\to_L)$$

$$\frac{\Gamma \vdash t : A \quad \Gamma; A \vdash k : B}{\Gamma \vdash tk : B}\ (\mathrm{Cut}) \qquad \frac{\Gamma, x : A \vdash v : B}{\Gamma; A \vdash [x]v : B}\ (\mathrm{Sel})$$

Proposition 3.1 (Admissible rule).
(i) *If* $\Gamma, x : A \vdash t : C$, *then* $\Gamma, x : A \cap B \vdash t : C$.
(ii) *If* $\Gamma, x : A; D \vdash k : C$, *then* $\Gamma, x : A \cap B; D \vdash k : C$.

Proof. By mutual induction on the structure of t and k. □

Proposition 3.2 (Basis expansion).
(i) $\Gamma \vdash t : A \;\Leftrightarrow\; \Gamma, x : B \vdash t : A$ *and* $x \notin \mathrm{Fv}(t)$.
(ii) $\Gamma; C \vdash k : A \;\Leftrightarrow\; \Gamma, x : B; C \vdash k : A$ *and* $x \notin \mathrm{Fv}(k)$.

Definition 3.3.

$$\begin{aligned}\Gamma_1 \cap \Gamma_2 = &\{x : A \mid x : A \in \Gamma_1 \ \&\ x \notin \Gamma_2\}\\ &\cup \{x : A \mid x : A \in \Gamma_2 \ \&\ x \notin \Gamma_1\}\\ &\cup \{x : A \cap B \mid x : A \in \Gamma_1 \ \&\ x : B \in \Gamma_2\}.\end{aligned}$$

Proposition 3.3 (Bases intersection). (i) $\Gamma_1 \vdash t : A \;\Rightarrow\; \Gamma_1 \cap \Gamma_2 \vdash t : A$.
(ii) $\Gamma_1; B \vdash k : A \;\Rightarrow\; \Gamma_1 \cap \Gamma_2; B \vdash k : A$.

Proposition 3.4 (Generation Lemma – GL).

(i) $\Gamma \vdash x : A$ *if and only if* $x : A \bigcap A_i \in \Gamma$, $i = 1, \dots, n$ *for some* $n \geqslant 0$.

(ii) $\Gamma \vdash \lambda x.t : A$ *if and only if* $A \equiv \bigcap B_i \to C$ *and* $\Gamma, x : \bigcap B_i \vdash t : C$.

(iii) $\Gamma; A \vdash [x]t : B$ *if and only if* $\Gamma, x : A \vdash t : B$.

(iv) $\Gamma \vdash tk : A$ *if and only if there is a type* B *such that* $\Gamma \vdash t : B$ *and* $\Gamma; B \vdash k : A$.

(v) $\Gamma; T \vdash t :: k : C$ *if and only if* $T \equiv \bigcap A_i \to B$, *and* $\Gamma; B \vdash k : C$ *and for all* i, $\Gamma \vdash t : A_i$.

Example 3.1. In λ-calculus with intersection types, the term $\lambda x.xx$ has the type $(A \cap (A \to B)) \to B$. The corresponding term in λ^{Gtz}-calculus is $\lambda x.x(x :: [y]y)$. Although being a normal form this term is not typeable in the simply typed λ^{Gtz}-calculus. It is typeable in $\lambda^{\mathsf{Gtz}}\cap$ in the following way:

$$\dfrac{\dfrac{\dfrac{}{x : A \cap (A \to B) \vdash x : A \to B}\,(\mathrm{Ax}) \qquad \dfrac{\dfrac{}{x : A \cap (A \to B) \vdash x : A}\,(\mathrm{Ax}) \quad \dfrac{\dfrac{}{x : A \cap (A \to B), y : B \vdash y : B}\,(\mathrm{Ax})}{x : A \cap (A \to B); B \vdash [y]y : B}\,(\mathrm{Sel})}{x : A \cap (A \to B); A \to B \vdash x :: [y]y : B}\,(\to_L)}{x : A \cap (A \to B) \vdash x(x :: [y]y) : B}\,(\mathrm{Cut})}{\vdash \lambda x.x(x :: [y]y) : (A \cap (A \to B)) \to B}\,(\to_R).$$

4. Subject reduction

Now, in order to prove preservation of types under reductions, so called Subject reduction property, we need to examine how meta-operators $\langle := \rangle$ and $@$ behave under reductions.

Lemma 4.1 (Substitution lemma).

(i) *If* $\Gamma, x : \bigcap A_i \vdash t : B$ *and* $\Gamma \vdash u : A_i$, *for each* i, *then* $\Gamma \vdash t\langle x := u\rangle : B$.

(ii) *If* $\Gamma, x : \bigcap A_i; C \vdash k : B$ *and* $\Gamma \vdash u : A_i$, *for each* i, *then* $\Gamma \vdash k\langle x := u\rangle : B$.

Proof. By induction on the structure of a term or a context.

• Basic case. 1. $t \equiv x$. In this case $x\langle x := u\rangle = u$. Then $\Gamma, x : \bigcap A_i \vdash x : B$, by GL(i), implies that $B \equiv A_i$, for some i, so we are done by the second premise.

2. $t \equiv y$. In this case $y\langle x := u\rangle = y$, so from $\Gamma, x : A \vdash y : B$ and Proposition 3.2 we have that $\Gamma \vdash y : B$.

• $t \equiv \lambda y.t'$ From $\Gamma, x : \bigcap A_i \vdash \lambda y.t' : B$ and by GL(ii) we get that $B \equiv \bigcap C_j \to D$ and for some j $\Gamma, x : \bigcap A_i, y : C_j \vdash t' : D$. In the case of t' by IH we get $\Gamma, y : C_j \vdash t'\langle x := u\rangle : D$ for all i. Since $(\lambda y.t')\langle x := u\rangle = \lambda y.t'\langle x := u\rangle$, we are done by Proposition 3.1 and $\to_R$.

• $t \equiv t'k$ $\Gamma, x : \bigcap A_i \vdash t'k : B$, using GL(iv), yields that there exists a type C, for which $\Gamma, x : \bigcap A_i \vdash t' : C$ and $\Gamma, x : \bigcap A_i; C \vdash k : B$. Using IH for both t' and k, we get:

$$\dfrac{\Gamma \vdash t'\langle x := u\rangle : C \quad \Gamma; C \vdash k\langle x := u\rangle : B}{\Gamma \vdash t'\langle x := u\rangle k\langle x := u\rangle : B,}$$

what we had to prove, since $(t'k)\langle x := u\rangle = t'\langle x := u\rangle k\langle x := u\rangle$.

• $k \equiv [y]v$ $\Gamma, x : \bigcap A_i; C \vdash [y]v : B$, by GL(iii), yields $\Gamma, x : \bigcap A_i, y : C \vdash v : B$. By IH applied to v, we get:

$$\frac{\Gamma, y : C \vdash v\langle x := u\rangle : B}{\Gamma; C \vdash [y]v\langle x := u\rangle : B,}$$

and that is what we needed, since $([y]v)\langle x := u\rangle = [y]v\langle x := u\rangle$.

• $k \equiv t :: k'$ From $\Gamma, x : \bigcap A_i; C \vdash t :: k' : B$ and GL(v) we have that $C \equiv \bigcap D_j \to E$, and $\Gamma, x : \bigcap A_i; E \vdash k' : B$ and $\Gamma, x : \bigcap A_i \vdash t : D_j$ for each j. By IH applied both to t and k' we get:

$$\frac{\Gamma \vdash t\langle x := u\rangle : D_j,\ \forall j \quad \Gamma; E \vdash k'\langle x := u\rangle : B}{\Gamma; \bigcap D_j \to E \vdash t\langle x := u\rangle :: k'\langle x := u\rangle : B,}$$

and since $(t :: k')\langle x := u\rangle = t\langle x := u\rangle :: k'\langle x := u\rangle$, the proof is completed. □

Lemma 4.2 (Append lemma). *If* $\Gamma; C \vdash k : B$ *and* $\Gamma; B \vdash k' : A$, *then* $\Gamma; C \vdash k@k' : A$.

Proof. By induction on the structure of the context.

• Basic case of the context structure is selection. So, if $k \equiv [x]v$ from $\Gamma; C \vdash [x]v : B$ and GL(iii) we have that $\Gamma, x : C \vdash v : B$. Without loss of generality we can suppose that $x \notin \mathrm{Fv}(k')$ (because if it was free in it, we would have to rename this variable in k where it is bound) and then we can expand the basis to $\Gamma, x : C; B \vdash k' : A$. Now

$$\cfrac{\cfrac{\Gamma, x : C \vdash v : B \quad \Gamma, x : C; B \vdash k' : A}{\Gamma, x : C \vdash vk' : A}}{\Gamma; C \vdash [x]vk' : A.}$$

Since $([x]v)@k' = [x]vk'$, the proof is completed.

• $k \equiv v :: k''$ $\Gamma; C \vdash v :: k'' : B$, by GL(v), yields that $C \equiv \bigcap D_i \to E$, $\Gamma \vdash v : D_i$ and $\Gamma; E \vdash k'' : B$. By applying IH to k'' we get:

$$\frac{\Gamma \vdash v : \bigcap D_i \quad \Gamma; E \vdash k''@k' : A}{\Gamma; C \vdash v :: (k''@k') : A.}$$

Since $(v :: k'')@k' = v :: (k''@k')$ the proof is completed. □

We are now able to prove Subject reduction theorem, which claims that the type of a term stays preserved under reduction.

Theorem 4.1 (Subject Reduction). *If* $\Gamma \vdash t : A$ *and* $t \to t'$, *then* $\Gamma \vdash t' : A$.

Proof. We examine three different cases, according to the last applied reduction.

• (β): If $\Gamma \vdash (\lambda x.t)(u :: k) : A$, then we have to show that $\Gamma \vdash t\langle x := u\rangle k : A$.

From $\Gamma \vdash (\lambda x.t)(u :: k) : A$ and GL(iv) we have that there is the type B such that $\Gamma \vdash \lambda x.t : B$ and $\Gamma; B \vdash u :: k : A$. GL(v) implies that $B \equiv \bigcap C_i \to D$, $\Gamma \vdash u : C_i$ and $\Gamma; D \vdash k : A$. On the other hand, from $\Gamma \vdash \lambda x.t : \bigcap C_i \to D$ we have from GL(ii) that $\Gamma, x : \bigcap C_i \vdash t : D$. Applying Substitution lemma 4.1, we get that $\Gamma \vdash t\langle x := u\rangle : D$, so we are now done by (Cut) rule.

• (σ): If $\Gamma \vdash t[x]v : A$, it should be shown that $\Gamma \vdash v\langle x := t\rangle : A$.

From $\Gamma \vdash t[x]v : A$ and GL(iv) it follows that there exists the type B such that $\Gamma \vdash t : B$ and $\Gamma; B \vdash [x]v : A$. Further, by GL(iii) we have that $\Gamma, x : B \vdash v : A$. Now, all assumptions of Lemma 4.1 are accomplished, so by applying it we get $\Gamma \vdash v\langle x := t\rangle : A$.

• (π): If $\Gamma \vdash (tk)k' : A$, we have to show that $\Gamma \vdash t(k@k') : A$.

From $\Gamma \vdash (tk)k' : A$ and GL(iv) we have that there is a type B such that $\Gamma \vdash tk : B$ and $\Gamma; B \vdash k' : A$. Further, $\Gamma \vdash tk : B$, by GL(iv), yields that there is the type C such that $\Gamma \vdash t : C$ and $\Gamma; C \vdash k : B$. Now from $\Gamma; C \vdash k : B$ and $\Gamma; B \vdash k' : A$, using Lemma 4.2 we get $\Gamma; C \vdash k@k' : A$, so we may conclude:

$$\frac{\Gamma \vdash t : C \quad \Gamma; C \vdash k@k' : A}{\Gamma \vdash t(k@k') : A.}$$ □

The reduction μ is of different nature, since it reduces contexts instead of terms. But it is possible to prove a similar result for this reduction rule.

Proposition 4.1. *If $\Gamma; \bigcap B_i \vdash [x]xk : A$, then $\Gamma; B_i \vdash k : A$, for some i.*

Proof. Assume $\Gamma; \bigcap B_i \vdash [x]xk : A$. By GL(iv) we have $\Gamma, x : \bigcap B_i \vdash xk : A$. Now, by GL(iii), there is a type C such that $\Gamma, x : \bigcap B_i \vdash x : C$ and $\Gamma; C \vdash k : A$. Since $x \notin k$, we are done by (Ax) and Proposition 3.2. □

5. Conclusion

We introduced an intersection type assignment system to an extension of the λ-calculus which corresponds to sequent calculus for intuitionistic implicational logic. For this system, we proved Subject reduction theorem. Our main goal, characterization of strongly normalizing terms via intersection types, is still in the domain of future work, as well as proving the confluence property for the system.

References

[1] H. Barendregt and S. Ghilezan, *Lambda terms for natural deduction, sequent calculus and cut elimination*, J. Funct. Program. 10:1 (2000), 121–134.

[2] M. Coppo and M. and Dezani-Ciancaglini, *A new type-assignment for lambda terms*, Arch. Math. Logik 19 (1978), 139–156.

[3] D. Dougherty, S. Ghilezan and P. Lescanne, *Characterizing strong normalization in the Curien–Herbelin symmetric lambda calculus: extending the Coppo–Dezani heritage*, Theoret. Comput. Sci. (2007), to appear

[4] R. Dyckhoff and L. Pinto, *Cut-Elimination and a Permutation-Free Sequent Calculus for Intuitionistic Logic*, Stud. Log. 60:1 (1998), 107–118.

[5] J. Espírito Santo and L. Pinto, *Permutative Conversions in Intuitionistic Multiary Sequent Calculi with Cuts*; in: *Procedings of TLCA 2003*, Lect. Notes Comput. Sci. 2071, 2003, 286–300.

[6] J. Espírito Santo, *Private communication*, 2006.

[7] S. Ghilezan, *Strong normalization and typability with intersection types*, Notre Dame J. Formal Logic 37 (1996), 44–52.

[8] J.-Y. Girard, *A New Constructive Logic: Classical Logic*, Math. Struct. Comput. Sci. 1:3 (1991), 255–296.

[9] H. Herbelin, *A lambda calculus structure isomorphic to Gentzen-style sequent calculus structure*; in: *Computer Science Logic, CSL 1994*, Lect. Notes Comput. Sci. 933, Springer-Verlag, 1995, 61–75.

[10] J. L. Krivine, *Lambda-calcul, types et modèles*, Masson, Paris, 1990.

[11] S. Lengrand, P. Lescanne, D. Dougherty, M. Dezani-Ciancaglini, and S. van Bakel, *Intersection types for explicit substitutions*, Inf. Comput. 189:1 (2004), 17–42.

[12] R. Matthes, *Characterizing Strongly Normalizing Terms of a Calculus with Generalized Applications via Intersection Types*; in: *ICALP Satellite Workshops 2000*, 339–354.

[13] G. Pottinger, *A type assignment for the strongly normalizable λ-terms*; in: J. P. Seldin and J. R. Hindley (eds.) *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, Academic Press, London, 1980, pp. 561–577

Fakultet tehnickih nauka
Univerzitet u Novom Sadu
Trg Dositeja Obradovica 6
21000 Novi Sad
Serbia
gsilvia@uns.ns.ac.yu
jelena@imft.ftn.ns.ac.yu