BULLETIN OF THE
GREEK MATHEMATICAL SOCIETY
Volume 55, 2008 (39–47)

Type Assignment and Conservation Properties

George Koletsos

Abstract

We use the type assignment method to prove conservation properties. First we give a type assignment proof of the classical conservation theorem for λI and then we extend this method to the notion of the reduction β_I and β_S. We also give a direct type assignment proof of the extended conservation property according to which if a term is β_I, β_S-normalizable then it is β-strongly normalizable.

1. Introduction

The classical conservation theorem for λI says that when a λI-term is β-normalizable then it is also β-strongly normalizable. The proofs of this theorem (see for example [2]) use a machinery of weight assignment to the terms where the weights are numbers and where the process of reduction decreases the weight of the term. As a result we cannot go for ever, and consequently we get strong normalization. In [6] a structural notion of reduction β_S is introduced, which along with β_I is used to prove strong normalization properties by first proving weak normalization properties. All the proofs used there exploit a variant of the weight assignment method, the labels à la Lévy (see [2]); a Church-Rosser theorem for labeled terms is also needed.

In this paper we take a different approach. We try to exploit the ideas and the results of the type assignment method. The idea of assigning types to terms is a quite old idea for foundational and programming purposes. By extending the original system of simple types to the system with intersection types it is possible to get characterizations of different classes of λ-terms (e.g. solvable or strongly normalizable, see [9] and [1]) and also to get proofs of important theorems of λ-calculus (see for example [9], [8]). So the type assignment method can be used as a proof tool for examining various properties of the λ-calculus. We try to exploit this idea in examining conservation properties.

2. Basic definitions

Here we remind the reader of some basic notions about the type-free λ-calculus.

Definition 2.1 The set Λ of λ-terms is inductively defined as follows: ($\mathcal{V} = \{x, y, z, \ldots\}$ is an infinite denumerable set of variables)

1 $x \in \mathcal{V} \Rightarrow x \in \Lambda$

2 $x \in \mathcal{V}, M \in \Lambda \Rightarrow (\lambda x.M) \in \Lambda$

3 $M, N \in \Lambda \Rightarrow (MN) \in \Lambda$

λ is a binding operator and the notions of free and bound occurrences of a variable in a term are defined as usual. $FV(M)$ denotes the set of free variables in the term M. Outer parentheses are usually omitted. We denote by $M_1 M_2 \cdots M_n$ the term $(\cdots((M_1 M_2) M_3) \cdots M_n)$. We also consider that some variable convention prevents us from clashes of variables. (Barendregt' s variable convention, see [2] or [3]).

The notion of substitution is also defined as usual.

$M[N_1/x_1, \ldots, N_k/x_k]$ denotes the result of simultaneously substituting the λ-terms $N_1, \ldots, N_k$ correspondingly for the free occurrences of the variables $x_1, \ldots, x_k$ in the λ-term M. The variable convention prevents the capture of free variables in N_i by any binding operator in M.

The principal notion of reduction of the λ-calculus is the notion of reduction β.

Definition 2.2 The notion of reduction β is defined by the following contraction rule

$$\beta : (\lambda x.M)N \rightarrow M[N/x]$$

The term $(\lambda x.M)N$ is called a *β-redex* and the term $M[N/x]$ its *contractum.* More generally

Definition 2.3 Any binary relation $R \subseteq \Lambda \times \Lambda$ is called a *notion of reduction.* In that case if $(M, N) \in R$ then M is called a *R-redex* and N is called the *contractum* of M (we write $R : M \rightarrow N$).

For R a notion of reduction we define the following relations between λ-terms:

— $\rightarrow_R$ is the least relation containing R and compatible with the λ-term formation rules, i.e. when we have $M \rightarrow_R N$ we also have $\lambda x.M \rightarrow_R \lambda x.N$, $(MO) \rightarrow_R (NO)$ and $(OM) \rightarrow_R (ON)$. It is easily proved that $M \rightarrow_R N$ if and only if exactly one R-redex in M is replaced by its contractum.

— $\twoheadrightarrow_R$ is the transitive, reflexive closure of the relation R.

We say also that a λ-term M is a R-normal form when no subterm of M is a R-redex i.e. $\nexists N$ s.t. $M \rightarrow_R N$. M is called R-normalizable if there exists a R-normal form N s.t. $M \twoheadrightarrow_R N$ and R - strongly normalizable if there is no infinite reduction sequence $M \equiv M_0 \rightarrow_R M_1 \rightarrow_R \cdots \rightarrow_R M_n \rightarrow_R \cdots$ starting from M.

The notion of β-redex can be split in two different notions of redexes.

Definition 2.4 Two notions of reduction β_I and β_K are defined by the following rules

$$\beta_I : (\lambda x.M)N \rightarrow M[N/x] \quad \text{if } \ x \in \mathrm{FV}(M)$$

$$\beta_K : (\lambda x.M)N \rightarrow M \quad \text{if } \ x \notin \mathrm{FV}(M)$$

Note that $\beta = \beta_I \cup \beta_K$ because in the case of β_K we have $M[N/x] = M$ as $x \notin FV(M)$. In contracting a β_K-redex the term N is thrown away.

Definition 2.5 We define a restricted class of λ-terms, the subset $\Lambda_I \subseteq \Lambda$ by the following inductive definition

1 $x \in \mathcal{V} \Rightarrow x \in \Lambda_I$

2 $M \in \Lambda_I$ and $x \in FV(M) \Rightarrow (\lambda x.M) \in \Lambda_I$

3 $M, N \in \Lambda_I \Rightarrow (MN) \in \Lambda_I$

λI-calculus is the λ-calculus restricted to Λ_I. Note that Λ_I is closed under the β-reduction relation. In fact a term $M \in \Lambda_I$ can have only β_I-redexes and if $M \rightarrow_\beta N$ then the redex contracted to get N is a β_I-redex and $N \in \Lambda_I$.

3. Intersection types. The system D

We introduce a system of intersection types, the system D.

Definition 3.1 *Types* are constructed from the variables $\alpha_1, \alpha_2, \ldots$ (a denumerable set of type variables) using the connectives $\rightarrow$ and $\wedge$.

We use $\sigma, \tau, \rho, \ldots$ to denote types. We denote by $\sigma_1 \rightarrow \sigma_2 \rightarrow \cdots \rightarrow \sigma_n \rightarrow \sigma$ the type $\sigma_1 \rightarrow (\sigma_2 \rightarrow (\cdots(\sigma_n \rightarrow \sigma)\cdots)$.

Definition 3.2 A *context* Γ is a set $\{x_1 : \sigma_1, \ldots, x_k : \sigma_k\}$ of variable assignments or declarations ($x_i \in \Lambda$ is a term variable and σ_i is a type). All the $x_1, \ldots, x_k$ in a context are distinct variables and if we write $\Gamma, x : \sigma$ we presuppose that x does not occur in Γ.

Definition 3.3 We define the notion $\Gamma \vdash M : \sigma$ (the subject M is typed with type σ in the context Γ) by the following rules:

1 $\Gamma, x : \sigma \vdash x : \sigma$ (axiom)

2 $\Gamma, x : \sigma \vdash M : \tau \Rightarrow \Gamma \vdash \lambda x.M : \sigma \rightarrow \tau$

3 $\Gamma \vdash M : \sigma \rightarrow \tau,\ \Gamma \vdash N : \sigma \Rightarrow \Gamma \vdash MN : \tau$

4 $\Gamma \vdash M : \sigma \wedge \tau \Rightarrow \Gamma \vdash M : \sigma$ and $\Gamma \vdash M : \tau$

5 $\Gamma \vdash M : \sigma$ and $\Gamma \vdash M : \tau \Rightarrow \Gamma \vdash M : \sigma \wedge \tau$

The system D is a system of *type assignment* or *typing system.* Every typing can be thought as a proof tree starting from the axioms 1 and using the inference rules 2–5. The following is immediate.

Lemma 3.4 1 If $\Gamma \vdash M : \sigma$ then $\Gamma, x : \tau \vdash M : \sigma$ (weakening)

2 If $\Gamma_0 \vdash M : \sigma$ and $\Gamma_1 \vdash N : \tau$ then there is a Γ s.t. $\Gamma \vdash M : \sigma$ and $\Gamma \vdash N : \tau$ (unification of contexts)

Proof: For the second case it suffices to take

$$\Gamma = \Gamma_0 \wedge \Gamma_1 = \{x : \tau \mid x : \tau \in \Gamma_i \text{ and } x \text{ does not occur in } \Gamma_{1-i}\} \cup \{x : \tau_0 \wedge \tau_1 \mid x : \tau_i \in \Gamma_i\}$$

and prove by induction the desired property.

Definition 3.5 A *prime type* is any type which is not an intersection i.e. it is either a type variable or a type of the form $\sigma \rightarrow \tau$.

Every type σ is an intersection of prime types called the *prime factors* of σ.

Lemma 3.6 (Generation lemma) Let $\Gamma \vdash M : \sigma$ with σ a prime type. Then

1 If M is a variable x then there is an assignment $x : \sigma'$ in Γ where σ is a prime factor of σ'.

2 If $M = \lambda x.N$ then $\sigma = \tau \to \rho$ and $\Gamma, x : \tau \vdash N : \rho$

3 If $M = NP$, then for some τ, $\Gamma \vdash P : \tau$ and $\Gamma \vdash N : \tau \to \sigma'$ where σ is a prime factor of σ'.

Proof: In the typing $\Gamma \vdash M : \sigma$ we choose a highest node $\Gamma \vdash M : \sigma'$ where σ is a prime factor of σ'. This node cannot be produced by using any of the rules 4 or 5. So we are left only with the rules 1 - 3 and the result follows then easily by inspection.

Definition 3.7 Let R be a notion of reduction.The typing system satisfies the *subject reduction* property for R if whenever $\Gamma \vdash M : \sigma$ and $M \to_R N$ imply that $\Gamma \vdash N : \sigma$. It satisfies the *subject expansion* property for R if $M \to_R N$ and $\Gamma \vdash N : \sigma$ imply $\Gamma \vdash M : \sigma$.

The subject reduction property holds for β. This is proved by induction on the way we get $\Gamma \vdash M : \sigma$ using the following proposition:

Proposition 3.8 Let Γ be a context and let $x_1, \ldots, x_k$ distinct variables not declared in Γ. If $\Gamma, x_1 : \sigma_1, \ldots, x_k : x_k \vdash M : \tau$ and if $\Gamma \vdash N_i : \sigma_i$ for every i $(1 \leq i \leq k)$ for which x_i is free in M, then $\Gamma \vdash M[N_1/x_1, \ldots, N_k/x_k] : \tau$.

In particular if $x_1, \ldots, x_k \notin FV(M)$ then $\Gamma, x_1 : \sigma_1, \ldots, x_k : \sigma_k \vdash M : \tau$ implies $\Gamma \vdash M : \tau$.

Proof: By induction on the way we get $\Gamma, x_1 : \sigma_1, \ldots, x_k : x_k \vdash M : \tau$ (see [9]).

Definition 3.9 A term M is *typable* (in D) if there exist Γ and σ s.t. $\Gamma \vdash M : \sigma$. It is *typable in the context* Γ if there exists σ s.t. $\Gamma \vdash M : \sigma$.

4. Type assignment and Conservation properties

Subject expansion, on the contrary, does not hold for β. The reason is that β-contraction (more specifically β_K-contraction) may throw away some of the terms. So we might be at a position to be able to type the contractum $M[N/x]$ of a β_K-redex $(\lambda x.M)N$ but not the redex itself. Consider for example the case when N cannot be typed (but N is not present in $M[N/x]$ as $M[N/x] = N$) while the typing of $(\lambda x.M)N$ requires a typing of N.

It turns out that if we are restricted only to β_I-redexes the subject expansion does hold. This is based on the following proposition.

Proposition 4.1 Let Γ be a context and let $x_1, \ldots, x_k$ variables not declared in Γ. If $\Gamma \vdash M[N_1/x_1, \ldots, N_k/x_k] : \tau$ and if the terms $N_1, \ldots, N_k$ are typable in the context Γ then there exist types $\sigma_1, \ldots, \sigma_k$ s.t. $\Gamma \vdash N_i : \sigma_i$ $(1 \leq i \leq k)$ and $\Gamma, x_1 : \sigma_1, \ldots, x_k : \sigma_k \vdash N : \tau$. In addition, if the variable x_i occurs free in M $(x_i \in FV(M))$ then N_i is typable in the context Γ.

Proof: Proof by induction on the structure of M and for a given M on the length of τ.

If $\tau = \tau' \wedge \tau''$ then $\Gamma \vdash M[N_1/x_1, \ldots, N_k/x_k] : \tau'$ and $\Gamma \vdash M[N_1/x_1, \ldots, N_k/x_k] : \tau''$. By I.H. (for τ' and τ'') there exist σ_i', σ_i'' s.t. $\Gamma \vdash N_i : \sigma_i'$ and $\Gamma \vdash N_i : \sigma_i''$

$(1 \leq i \leq k)$ and $\Gamma, x_1 : \sigma'_1, \ldots, x_k : \sigma'_k \vdash N : \tau'$ and $\Gamma, x_1 : \sigma''_1, \ldots, x_k : \sigma''_k \vdash N : \tau''$. Putting $\sigma_i = \sigma'_i \wedge \sigma''_i$ we have that $\Gamma \vdash N_i : \sigma_i$ and $\Gamma, x_1 : \sigma_1, \ldots, x_k : \sigma_k \vdash N : \tau$ (lemma 3.4).

If, in addition, $x_i \in FV(M)$ then by I.H. N_i is typable in the context Γ.

So we can consider that τ is a prime type. We have the following possibilities.

1) M is a variable

If $M = x_i$ then $M[N_1/x_1, \ldots, N_k/x_k] = N_i$. So $\Gamma \vdash N_i : \tau$. So we can take $\sigma_i = \tau$ and for σ_j $(j \neq i)$ we can take any of the possible types for M_j (it exists by hypothesis). We then have $\Gamma, x_1 : \sigma_1, \ldots, x_k : \sigma_k \vdash x_i : \tau$. Of course if x_j occurs free in M then $x_j = x_i$ so N_i is typable because $\Gamma \vdash N_i : \tau$.

If $M \neq x_1, \ldots, x_k$ then $M[N_1/x_1, \ldots, N_k/x_k] = M$. So by hypothesis $\Gamma \vdash M : \tau$. We can take σ_i any of the possible types of N_i and have that $\Gamma, x_1 : \sigma_1, \ldots, x_k : \sigma_k \vdash N : \tau$.

The additional condition holds vacuously (for any x_i, $x_i \notin FV(M)$).

2) $M = \lambda y.P$. We have that $M[N_1/x_1, \ldots, N_k/x_k] = \lambda y.P[N_1/x_1, \ldots, N_k/x_k]$ is typed in the context Γ with the prime type τ. By lemma 3.6 $\tau = \rho \to \phi$ and $\Gamma, y : \rho \vdash P[N_1/x_1, \ldots, N_k/x_k] : \phi$. By I.H. there exist types σ_i s.t. $\Gamma, y : \rho \vdash N_i : \sigma_i$ and $\Gamma, y : \rho, x_1 : \sigma_1, \ldots, x_k : \sigma_k \vdash P : \tau$. As y is a bound variable we can suppose that $y \notin FV(N_i)$, so by proposition 3.8 we take $\Gamma \vdash N_i : \sigma_i$. We take also that $\Gamma, x_1 : \sigma_1, \ldots, x_k : \sigma_k \vdash \lambda y.P : \rho \to \phi$ (rule 2).

In addition if $x_i \in FV(M)$ then $x_i \in FV(P)$ and $x_i \neq y$. So from $\Gamma, y : \rho \vdash P[N_1/x_1, \ldots, N_k/x_k] : \phi$ and the I.H. $\Gamma, y : \rho \vdash N_i : \sigma_i$ (for some σ_i). But as $y \notin FV(N_i)$ we take $\Gamma \vdash N_i : \sigma_i$.

3) $M = (PQ)$. By hypothesis $\Gamma \vdash P[N_1/x_1, \ldots, N_k/x_k]Q[N_1/x_1, \ldots, N_k/x_k] : \tau$ and τ prime type. From lemma 3.6 we have that $\Gamma \vdash Q[N_1/x_1, \ldots, N_k/x_k] : \rho$ and $\Gamma \vdash P[N_1/x_1, \ldots, N_k/x_k] : \rho \to \tau'$ where τ is a prime factor of τ'. From I.H. there exist σ'_i, σ''_i such that $\Gamma \vdash N_i : \sigma'_i$ and $\Gamma, x_1 : \sigma'_1, \ldots, x_k : \sigma'_k \vdash Q : \rho$ $\Gamma \vdash N_i : \sigma''_i$ and $\Gamma, x_1 : \sigma''_1, \ldots, x_k : \sigma''_k \vdash P : \rho \to \tau'$.

If we put $\sigma_i = \sigma'_i \wedge \sigma''_i$ then $\Gamma \vdash N_i : \sigma_i$ and $\Gamma, x_1 : \sigma_1, \ldots, x_k : \sigma_k \vdash Q : \rho$ and $\Gamma, x_1 : \sigma_1, \ldots, x_k : \sigma_k \vdash P : \rho \to \tau'$ from which $\Gamma, x_1 : \sigma_1, \ldots, x_k : \sigma_k \vdash PQ : \tau'$ and by using rule 4 we finally get $\Gamma, x_1 : \sigma_1, \ldots, x_k : \sigma_k \vdash PQ : \tau$.

If in addition $x_i \in FV(M)$ then x_i must occur free in either P or Q, say Q. But from $\Gamma \vdash Q[N_i/x_1, \ldots, N_k/x_k] : \rho$ and I.H. N_i is typable in the context Γ.

We can now prove the main result.

Theorem 4.2 (Subject expansion) The typing system D satisfies the subject expansion property for β_I.

Proof: It suffices to prove that if $M \to_{\beta_I} N$ and $\Gamma \vdash N : \sigma$ then $\Gamma \vdash M : \sigma$. We prove this by induction on the structure of M and for a given M by induction on the length of σ.

If $\sigma = \sigma_1 \wedge \sigma_2$ then $\Gamma \vdash N : \sigma_1 \wedge \sigma_2$. So $\Gamma \vdash N : \sigma_1$ and $\Gamma \vdash N : \sigma_2$. By I.H. $\Gamma \vdash M : \sigma_1$ and $\Gamma \vdash M : \sigma_2$, from which $\Gamma \vdash M : \sigma$.

So we can consider that σ is a prime type. We have the following possibilities:

1) M is a variable. This case is impossible as $M \to_{\beta I} N$.

2) $M = \lambda x.P$. Then $P \to_{\beta_I} P'$ and $N = \lambda x.P'$. Because σ is prime we have $\sigma = \rho \to \tau$ and $\Gamma, x : \rho \vdash P' : \tau$. From I.H. we get $\Gamma, x : \rho \vdash P : \tau$ so $\Gamma \vdash \lambda x.P : \rho \to \tau$.

3) $M = (PQ)$. We have three possibilities:

3.1) $P \rightarrow_{\beta_I} P'$ and $N = P'Q$. Because σ is prime we have that $\Gamma \vdash P' : \tau \rightarrow \sigma'$ and $\Gamma \vdash Q : \tau$ where σ is a prime factor of σ'. By I.H. $\Gamma \vdash P : \tau \rightarrow \sigma'$ and then $\Gamma \vdash PQ : \sigma'$ from which $\Gamma \vdash PQ : \sigma$.

3.2) $Q \rightarrow_{\beta_I} Q'$ and $N = PQ'$. Same argument as in 3.1.

3.3) $P = \lambda x.W$. So $M = (\lambda x.W)Q$ and $N = W[Q/x]$. In this case M is a β_I-redex and $x \in FV(W)$. From proposition 4.1 and the fact that $\Gamma \vdash W[Q/x] : \sigma$ we have $\Gamma \vdash Q : \tau$ (for some τ) and $\Gamma, x : \tau \vdash V : \sigma$. So $\Gamma \vdash \lambda x.W : \tau \rightarrow \sigma$, so finally $\Gamma \vdash (\lambda x.W)Q : \sigma$.

How can we type normal forms in the type system D? It is well known that the β-normal forms can be typed in D. We present the proof and later on we' ll try to investigate under which conditions a more relaxed notion of normal form (β_I-normal form) can be typed in D.

Proposition 4.3 If M is β-normal then M can be typed in D. Moreover, if M is not an abstraction then M can be typed in D with any type σ.

Proof: By induction on M. If M is a variable is obvious. If $M = \lambda x.N$ then $\Gamma \vdash N : \tau$, by I.H. Either x is declared in Γ or x does not occur in Γ In either case (in the second by weakening) we have that $\Gamma, x : \rho \vdash N : \tau$ so $\Gamma \vdash \lambda x.N : \rho \rightarrow \tau$. If $M = PN$ then P is not an abstraction. By I.H. $\Gamma_1 \vdash N : \tau$ and $\Gamma_2 \vdash P : \tau \rightarrow \sigma$ (we can use the type $\tau \rightarrow \sigma$ because P is not an abstraction). But then we can easily get by $\Gamma = \Gamma_1 \wedge \Gamma_2$ that $\Gamma \vdash PN : \sigma$.

What if a term M is typed in the system D? The following holds.

Theorem 4.4 If $\Gamma \vdash M : \sigma$ then M is strongly normalizable.

Proof: The proof uses the reducibility method. This can be found in [9].

We are now able to give an alternative proof of the famous conservation theorem (Church [4]).

Theorem 4.5 (Conservation theorem) In the λI-calculus if M is normalizable then M is strongly normalizable.

Proof: Let M be normalizable. We have then that $M \twoheadrightarrow_\beta N$ and N is β-normal. But in the λI-calculus any β-redex is a β_I-redex and any β-reduction is in fact a β_I-reduction. So we have that $M \twoheadrightarrow_{\beta_I} N$ and by 4.3 $\Gamma \vdash N : \sigma$. By the Subject expansion theorem 4.2 $\Gamma \vdash M : \sigma$. The result follows using theorem 4.4.

It is not true that all β_I-normal forms can be typed in D. The reason is that a contraction of a β_K-redex (in β_I-normal form) can create a β_I-redex which is not normalizable e.g.

$$(\lambda y.(\lambda x.xx))N(\lambda x.xx) \rightarrow_K (\lambda x.xx)(\lambda x.xx)$$

We see that the term at the left has the form $(\lambda x.M)NO$ where $x \notin FV(M)$. What if we didn't allow such a term to occur? Following [6] we call any such term a β_S-redex and define a notion of reduction

$$\beta_S : (\lambda x.M)NO \rightarrow (\lambda x.MO)N \quad (\text{when} \;\; x \notin FV(M))$$

Remark 4.6 Because of the variable convention the bound variable $x \notin FV(MO)$.

Lemma 4.7 If the λ-term M is not an abstraction then M is uniquely written as $NM_1 \cdots M_n$ where N is either a variable or an abstraction i.e. either $M = xM_1 \cdots M_n$ or $M = (\lambda x.P)M_1 \cdots M_n$ (in the case M is a variable we have the first case with $n = 0$ while in the second case we always have $n \geqslant 1$). N is called the *head* of M.

Proof: Easy induction on M.

Lemma 4.8 If the terms $M_1, \ldots, M_n$ are typable in D then the term $xM_1 \cdots M_n$ is typable in D.

Proof: By hypothesis $\Gamma_1 \vdash M_1 : \sigma_1$,.., $\Gamma_n \vdash M_n : \sigma_n$. The variable x is either declared in Γ_i with type τ_i or else we can add the declaration $x : \tau_i$ and replace Γ_i by $\Gamma_i, x : \tau_i$. So x is declared in $\Gamma = \Gamma_1 \wedge \cdots \wedge \Gamma_n$ with type $\tau = \tau_1 \wedge \cdots \wedge \tau_n$ and $\Gamma \vdash M_i : \sigma_i$ (lemma 3.4). Replace τ by $\tau \wedge (\sigma_1 \to \cdots \to \sigma_n \to \sigma)$ and get that with the new context Γ' we have that $\Gamma' \vdash x : \sigma_1 \to \sigma_2 \to \cdots \to \sigma_n \to \sigma$ and $\Gamma' \vdash M_i : \sigma_i$. By repeatedly using rule 3 we get $\Gamma' \vdash xM_1 \cdots M_n : \sigma$.

We can now prove the following proposition.

Proposition 4.9 Let M be in β_I, β_S-normal form i.e. it does not contain any β_I or any β_S redex. Then M can be typed in D.

Proof: By induction on the structure of M. We have the following cases:

M is a variable. The case is obvious.

$M = \lambda x.P$. Then by I.H. $\Gamma \vdash P : \sigma$ and x is either already declared in Γ or else can be added by weakening i.e. $\Gamma, x : \tau \vdash P : \sigma$ from which $\Gamma \vdash \lambda x.P : \tau \to \sigma$.

M is a composition. We have two cases (lemma 4.7).

Either $M = xM_1 \cdots M_n$ $(n \geqslant 1)$ which is typed by using I.H. and 4.8.

or $M = (\lambda x.P)M_1 \cdots M_n$ $(n \geqslant 1)$. We have that $x \notin FV(P)$ because otherwise we would have a β_I-redex. And if $n > 1$ then we would have a β_S-redex. So $M = (\lambda x.P)M_1$ and by I.H. $\Gamma_1 \vdash M_1 : \tau$ and $\Gamma_2 \vdash P : \sigma$. So $\Gamma \vdash M_1 : \tau$ and $\Gamma \vdash P : \sigma$. Because $x \notin FV(P)$, by lemma 3.8 we can consider that x is not declared in Γ. By variable convention $x \notin FV(M_1)$, so x can be considered not declared in Γ either (that is in unifying Γ_1 and Γ_2 there is no need to an assignment for x). Therefore by weakening $\Gamma, x : \tau \vdash M_1 : \sigma$ which gives $\Gamma \vdash \lambda x.P : \tau \to \sigma$ which combined with $\Gamma \vdash M_1 : \tau$ gives $\Gamma \vdash (\lambda x.P)M_1 : \sigma$.

We have proved that the β_I, β_S-normal form term can be typed in D. If we could reach, starting from M, a β_I, β_S-normal form of M then provided that this procedure satisfies the subject expansion property we would get as a result that M could be typed in D. It turns out that this is achieved with the β_S-reduction.

Proposition 4.10 If $M \to_{\beta_S} N$ and $\Gamma \vdash N : \sigma$ then $\Gamma \vdash M : \sigma$.

Proof: We prove the proposition by induction on the structure of M, and for a given M by induction on the length of σ. The case $\sigma = \sigma_1 \wedge \sigma_2$ is trivial and as in theorem 4.2 we can consider that σ is a prime type. Exactly as in 4.2 we treat the different cases where M is either a variable or $M = \lambda x.P$ with $P \to_{\beta_S} P'$ and $N = \lambda x.P'$ or $M = PQ$ and $P \to_{\beta_S} P'$, in which case $N = P'Q$ (alternatively $Q \to_{\beta_S} Q'$

in which case $N = PQ'$). The only interesting new case is when $M = (\lambda x.P)QO$ and $N = (\lambda x.PO)Q$ i.e. when the redex we contract is M.

By hypothesis $\Gamma \vdash (\lambda x.PO)Q : \sigma$. Because σ is prime by 3.6 we take $\Gamma \vdash (\lambda x.PO) : \tau \to \sigma'$ and $\Gamma \vdash Q : \tau$ where σ is prime factor of σ'. Still by lemma 3.6 $\Gamma, x : \tau \vdash PO : \sigma'$ from where $\Gamma, x : \tau \vdash P : \rho \to \sigma''$ and $\Gamma \vdash O : \rho$. By using the typing rules we take $\Gamma \vdash \lambda x.P : \tau \to (\rho \to \sigma'')$ and $\Gamma \vdash (\lambda x.P)Q : \rho \to \sigma'$ and $\Gamma \vdash (\lambda x.P)QO : \sigma''$ from which $\Gamma \vdash (\lambda x.P)QO : \sigma$.

Theorem 4.11 (Subject expansion for $\beta_I \cup \beta_S$) The typing system D satisfies the subject expansion property for $\beta_I \cup \beta_S$.

Proof: Immediate from 4.2 and 4.10.

We are now able to prove an extension of the classical conservation property.

Theorem 4.12 (Extended Conservation theorem) If the term M is β_I, β_S-normalizable then M is strongly β-normalizable.

Proof: By hypothesis $M \twoheadrightarrow_{\beta_I,\beta_S} N$ where N is β_I, β_S-normal. But then by 4.9 $\Gamma \vdash N : \sigma$ and by 4.11 $\Gamma \vdash M : \sigma$. Theorem 4.4 gives that M is strongly β-normalizable.

Conservation properties are useful in the sense that they permit easier strong normalization proofs. In general a proof of weak normalization (i.e. that a term has a normal form) is much easier than a proof of strong normalization. In fact many strong normalization proofs can be given by using conservation properties and proofs of weak normalization (for details see [6]). But even for checking that a term is strongly normalizable sometimes it suffices to check a weak normalization property. For example, the previous propositions give the following.

Proposition 4.13 If N is β_I, β_S-normal and $M \twoheadrightarrow_{\beta_I} N$ or $M \twoheadrightarrow_{\beta_S} N$ or $M \twoheadrightarrow_{\beta_I,\beta_S} N$ then M is strongly normalizable.

In 4.12 we proved an extension of a conservation property. A question arises whether a classical conservation property holds for β_I, β_S. That is, if a term M is weakly β_I, β_S-normalizable is it also β_I, β_S-strongly normalizable? We prove this by using the reducibility method. Some intermediate lemmas are proven first.

Lemma 4.14 If $M \twoheadrightarrow_{\beta_I,\beta_S} N$ then $FV(M) = FV(N)$.

Proof: We prove the property by induction on M in the case $M \to_{\beta_I,\beta_S} N$. The interesting cases are when $M = (\lambda x.P)Q,\ x \in FV(P)$ and $N = P[Q/x]$. In this case we see that the free variables $\neq x$ of P are retained in N and the free variables of Q are retained as well because Q is in fact a subterm of N ($x \in FV(P)$).

When $M = (\lambda x.P)QO$ and $x \notin FV(P)$ then it is obvious that this term has the same free variables with $(\lambda x.PO)Q$.

Lemma 4.15 If $M \to_{\beta_I,\beta_S} N$ then $P[M/x] \twoheadrightarrow_{\beta_I,\beta_S} P[N/x]$.

Proof: Induction on the structure of P.

Lemma 4.16 If $M \to_{\beta_I,\beta_S} N$ then $M[P/x] \to_{\beta_I,\beta_S} N[P/x]$.

Proof: By induction on the structure of M. From the definition of $\rightarrow_{\beta_I,\beta_S}$ we have the following possibilities:

$M = \lambda y.Q$, $Q \rightarrow_{\beta_I,\beta_S} Q'$ and $N = \lambda y.Q'$. Use I.H.

$M = QW$. If either $Q \rightarrow_{\beta_I,\beta_S} Q'$ or $W \rightarrow_{\beta_I,\beta_S} W'$, use I.H. So we are left with the case when N is produced by contracting $M = QW$, this being a redex. This can only happen if either M is a β_I-redex or M is a β_S-redex. So we have two cases.

Case 1. $Q = \lambda y.Q_1$ and $N = Q_1[W/y]$ and $y \in FV(Q_1)$. It suffices to prove that $((\lambda y.Q_1)W)[P/x] \rightarrow_{\beta_I} (Q_1[W/y])[P/x]$.
But $((\lambda y.Q_1)W)[P/x] = ((\lambda y.Q_1)[P/x])(W[P/x]) = (\lambda y.(Q_1[P/x]))(W[P/x]) \rightarrow_{\beta_I} (Q_1[P/x])[W[P/x]/y] = (Q_1[W/y])[P/x]$ (by substitution lemma in [3]).

Remark 4.17 The above contraction is indeed a β_I contraction because as $y \in FV(Q_1)$ it remains true that $y \in FV(Q_1[P/x])$. The bound variable y can be considered different from x.

Case 2. $Q = (\lambda y.Q_1)Q_2$ and $N = (\lambda y.Q_1 W)Q_2$ $(y \notin FV(Q_1))$. We have $(\lambda y.Q_1)Q_2W[P/x] = (\lambda y.Q_1[P/x])(Q_2[P/x])(W[P/x]) \rightarrow_{\beta_S}$
$(\lambda y.(Q_1[P/x])(W[P/x]))(Q_2[P/x]) = ((\lambda y.Q_1 W)Q_2)[P/x]$.

Remark 4.18 The redex above is a β_S-redex because $y \notin FV(Q_1[P/x])$. $y \notin FV(Q_1)$ by hypothesis and $y \notin FV(P)$ by variable convention.

Lemma 4.19 Let $M \twoheadrightarrow_{\beta_I,\beta_S} N$ and $P \twoheadrightarrow_{\beta_I,\beta_S} Q$. Then $M[P/x] \twoheadrightarrow_{\beta_I,\beta_S} N[Q/x]$.

Proof: From lemma 4.15 $M[P/x] \twoheadrightarrow_{\beta_I,\beta_S} N[P/x]$. From lemma 4.16 $N[P/x] \twoheadrightarrow_{\beta_I,\beta_S} N[Q/x]$. The result follows by transitivity.

Lemma 4.20 Let N and $MM_1 \cdots M_n$ $(n > 0)$ be β_I, β_S strongly normalizable. Then if $x \notin FV(M)$ the term $(\lambda x.M)NM_1 \cdots M_n$ is β_I, β_S strongly normalizable.

Proof: Induction on n.

$n = 0$. Then the term $(\lambda x.M)N$ is not a β_I-redex and it is not a β_S-redex either. So the only possible reducts of this term have the form $(\lambda x.M')N'$ where M' and N' are possible reducts of M and N. Note that the terms $(\lambda x.M')N'$ cannot be β_I-redexes (as by lemma 4.14 we have that $x \notin M'$ and evidently can neither be β_S-redexes). So $(\lambda x.M)N$ is strongly normalizable as M and N are strongly normalizable.

$n > 0$. We examine the term $(\lambda x.M)NM_1 \cdots M_n$. How is it possible to create an infinite β_I, β_S reduction starting from this term? If we start contracting redexes in $M, N, M_1, \ldots, M_n$ this procedure must terminate as all these terms are strongly normalizable. Therefore in order to create a possible infinite reduction we must contract a redex at the beginning of the term $(\lambda x.M')N'M_1' \cdots M_n'$. This is not a possible β_I-redex because $x \notin FV(M')$ (the β_I, β_S-reduction from M to M' cannot create new free variables, lemma 4.14). So eventually a β_S-redex of the form $(\lambda x.M')N'M_1'$ must be contracted to get $(\lambda x.M'M_1')N'M_2' \cdots M_n'$. But then N', $M'M_1'$ and $(M'M_1')M_2' \cdots M_n'$ are all strongly normalizable. Use I.H. to get the result.

Proposition 4.21 Let $\mathcal{P}$ be a property of the λ-terms i.e. $\mathcal{P} \subseteq \Lambda$. Then if $\Gamma \vdash M : \sigma$ we have that M has the property $\mathcal{P}$ i.e. $M \in \mathcal{P}$ provided $\mathcal{P}$ satisfies the following properties:

1 If $M_1, \ldots, M_n \in \mathcal{P}$ then $xM_1, \ldots, M_n \in \mathcal{P}$.

2 If $Mx \in \mathcal{P}$ then $M \in \mathcal{P}$.

3 $\mathcal{P}$ is $\mathcal{P}$-saturated that is if $N \in \mathcal{P}$ and $M[N/x]M_1 \cdots M_n \in \mathcal{P}$ then $(\lambda x.M)NM_1 \cdots M_n \in \mathcal{P}$.

Proof: See Krivine [9], pages 42 – 45. For a more general treatment of the method of reducibility see also Koletsos - Stavrinos [7] and Ghilezan and Likavec [5].

Theorem 4.22 Suppose $\Gamma \vdash M : \sigma$. Then M is β_I, β_S strongly normalizable.

Proof: According to proposition 4.21 it suffices to prove that the set $\mathcal{P} = \{M \mid M$ is β_I, β_S-strongly normalizable$\}$ satisfies the properties 1, 2 and 3 of this proposition.

That $\mathcal{P}$ satisfies the properties 1 and 2 is immediate. So in order to finish the proof we must prove that $\mathcal{P}$ is $\mathcal{P}$-saturated.

Let $N \in \mathcal{P}$ and $M[N/x]M_1 \cdots M_n \in \mathcal{P}$. We must prove that

$$(\lambda x.M)NM_1 \cdots M_n \in \mathcal{P}, \ \ (n \geq 0).$$

We prove it by induction on $p+q$ where p (resp. q) is the sum of the lengths of all the β_I, β_S-normalizations of N (resp. of $M[N/x]M_1 \cdots M_n$). We consider all the possible terms obtained by contracting a β_I, β_S-redex in $(\lambda x.M)NM_1 \cdots M_n$. It suffices to prove that are all in $\mathcal{P}$.

1. We contract the redex $(\lambda x.M)N)$ (this must be a β_I-redex). But the obtained term $M[N/x]M_1 \cdots M_n$ is in $\mathcal{P}$.

2. We have a possible β_S-redex $(\lambda x.M)NM_1$. But then $x \notin M$ and by lemma 4.20 the term $(\lambda x.M)NM_1 \cdots M_n$ is strongly normalizable.

3. We contract a redex in M and get M'. We must prove that $\lambda x.M')NM_1 \cdots M_n \in \mathcal{P}$. But from lemma 4.15 $M[N/x] \to M'[N/x]$. So the sum of all the normalizations of $M'[N/x]M_1 \cdots M_n$ is less than q and the result follows by I.H.

4. We contract a redex in M_i. Same proof.

5. We contract a redex in N and get N'. So the sum of normalizations of N' is less than p. On the other hand by lemma 4.16 $M[N/x] \to M[N'/x]$ so the sum of normalizations of $M[N'/x]M_1 \cdots M_n$ is less than the sum of normalizations of $M[N/x]M_1 \cdots M_n$. We use I.H.

We are now able to prove the conservation theorem for β_I, β_S-reduction.

Theorem 4.23 If M is β_I, β_S-normalizable then M is β_I, β_S-strongly normalizable.

Proof. The β_I, β_S-normal form of M is typable in D. By theorem 4.11 M also is typable in D and by 4.22 M is β_I, β_S-strongly normalizable.

Conclusion: We saw how the type assignment method can be used to organize and prove conservation properties for various notions of reduction in λ-calculus. The method is general, easy to handle and is based on subject expansion properties and typing characterizations of some classes of λ-terms. The extension of the method to other related notions of reduction would be the subject of future work.

References

1. S. van Bakel. Complete restrictions of the intersection type discipline. Theoretical Computer Science, 102 (1): 135 – 163, 1992.
2. H.P. Barendregt. The lambda calculus, its syntax and semantics. North-Holland, 1984.
3. H.P. Barendregt. Lambda calculi with types. Handbook of Logic in Computer science, Vol 2, Oxford 1992.
4. A. Church. The Calculi of Lambda Conversion, Princeton University Press, Princeton 1941.
5. S. Ghilezan and S. Likavec. Recucibility: A Ubiquitous Method in Lambda Calculus with Intersection Types. Electr. Notes Theor. Comput. Sci. 70 (1): (2002).
6. P. de Groote. The Conservation theorem revisited. Proceedings of the International Conference on Typed Lambda Calculi and Applications, Lecture Notes in Computer Science, Vol. 664, Springer-Verlag (1993) pp. 163–178.
7. G. Koletsos and G. Stavrinos. The structure of reducibility proofs. G. Koletsos, Ph. Kolaitis (eds.), Proceedings of the 2nd Panhellenic Logic Symposium, Delphi 1999, pp 138 – 143.
8. G. Koletsos and G. Stavrinos. Church-Rosser theorem for conjunctive type systems, in: A.K. Kakas, A. Sinachopoulos (eds.), Proceedings of the First Panhellenic Symposium on Logic, University of Cyprus (1997), pp 25 – 37.
9. J.L. Krivine. Lambda-calcul, types et modèles. Masson, 1990.

◇ George Koletsos
National Technical University of Athens
School of electrical and Computer Engineering
Department of Computer Science
Zografou Campus - 157 80 Athens - Greece
`koletsos@math.ntua.gr`