A STRONG NORMALISATION CONDITION FOR PURE TYPE SYSTEMS

MILENA STEFANOVA* AND SILVIO VALENTINI**

*Agency for sustainable growth, ENEA, via Don Fiammelli 2, I–40100 Bologna (Italy)
e-mail address: milena.stefanova@bologna.enea.it

**Dip. Matematica Pura e Applicata, Università di Padova, via Trieste 63, I–35121 Padova (Italy)
e-mail address: silvio@math.unipd.it

Abstract. A proof theoretical analysis suggests that the process of cut elimination in a sequent calculus corresponds to the normalisation of the proofs in natural deduction. If one moves to proofs decorated with lambda terms according to the Curry-Howard's isomorphism, the same observation leads to realising that the process of normalisation of the lambda-terms decorating the proofs in natural deduction is deeply connected with the property of closure under substitution of the lambda terms decorating the proofs in a sequent calculus. In this paper we show that this observation becomes a criterion to recognise the strongly normalising Pure Type Systems.

Keywords: Pure Type Systems, normalisation, proof theory.

1. Introduction

In this paper we deal with *Pure Type Systems* (PTSs for short). These typing systems provide a framework to describe in a uniform way a large class of type systems. PTSs were introduced by S. Berardi (see [3]) generalising the abstraction mechanism of the λ-cube of Barendregt (see [1]). Work of J. Terlouw contributed to the precise form of the rules for these systems (see [10]).

Each PTS is determined by its *specification*.

Definition 1.1 (PTS specification)**.** A *specification* for a PTS is a triple $\mathcal{S} \equiv \langle \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$, where S is a set whose elements are called *sorts*, A is a set of couples, called *axioms*, of the form (s_1, s_2) for $s_1, s_2 \in \mathsf{S}$, and R is a set of triples, called *rules*, of the form (s_1, s_2, s_3) for $s_1, s_2, s_3 \in \mathsf{S}$.

The intention of any axiom (s_1, s_2) is stating that the sort s_1 has type s_2 and the intention of any rule (s_1, s_2, s_3) is stating that the corresponding Π-formation rule holds (see below).

Definition 1.2 (PTS)**.** A *Pure Type System* is a couple $\lambda S = \langle \mathcal{S}, \vdash \rangle$, such that $\mathcal{S}$ is the specification of λS and $\vdash$ is a *judgement relation* determined by the following derivation rules:

REFLECTIONS ON TYPE THEORY, λ-CALCULUS, AND THE MIND
Essays Dedicated to Henk Barendregt on the Occasion of his 60th Birthday

(axioms)	$\vdash s_1 : s_2$	$(s_1, s_2) \in \mathsf{A}$
(variable)	$\dfrac{\Gamma \vdash T : s}{\Gamma, v{:}T \vdash v : T}$	$s \in \mathsf{S}\ ,\ v \notin FV(\Gamma)$
(weakening)	$\dfrac{\Gamma \vdash T : s \quad \Gamma \vdash M : U}{\Gamma, v{:}T \vdash M : U}$	$s \in \mathsf{S}\ ,\ v \notin FV(\Gamma)$
(Π-formation)	$\dfrac{\Gamma \vdash T : s_1 \quad \Gamma, v{:}T \vdash U : s_2}{\Gamma \vdash \Pi v : T.U : s_3}$	$(s_1, s_2, s_3) \in \mathsf{R}$
(λ-abstraction)	$\dfrac{\Gamma, v{:}T \vdash M : U \quad \Gamma \vdash \Pi v : T.U : s}{\Gamma \vdash (\lambda v : T.M) : \Pi v : T.U}$	$s \in \mathsf{S}$
(application)	$\dfrac{\Gamma \vdash M : \Pi v : T.U \quad \Gamma \vdash N : T}{\Gamma \vdash MN : U[v := N]}$	
(conversion)	$\dfrac{\Gamma \vdash M : T \quad \Gamma \vdash U : s}{\Gamma \vdash M : U}$	$T \sim U \quad s \in \mathsf{S}$

where by $T \sim U$ we mean that T and U are equal according to the minimal congruence relation induced by the β-reduction, namely, the computation process which states that $(\lambda v : T.M)N$ reduces to $M[v := N]$.

1.1. **Properties of PTSs.** In the next sections we will need to use some well known properties of PTSs, like for instance the *inversion lemma*, the *subject reduction lemma* and the property of *closure under substitution*. We give for granted that the reader is aware of them (see [1] or [6]). But, we will also use the notation that we introduce here. We will write $\Gamma \twoheadrightarrow \Gamma'$ (respectively, $\Gamma \sim \Gamma'$) to mean that Γ and Γ' are contexts of the same length which contain the same free variables and the types of these variables in Γ reduce (respectively, are equivalent) to the corresponding types in Γ'. An example of use of this notation is the following straightforward lemma.

Lemma 1.3. For any PTS λS the following properties hold:
(i) if $\Gamma \vdash M : T$ and $\Gamma \twoheadrightarrow \Gamma'$, then $\Gamma' \vdash M : T$;
(ii) if $\Gamma \vdash M : T$ and $T \twoheadrightarrow T'$, then $\Gamma \vdash M : T'$.

We will be concerned mainly with a special kind of PTSs, the so called *singly-sorted* PTSs which enjoy the property of type uniqueness, namely, if $\Gamma \vdash M : T$ and $\Gamma \vdash M : T'$ then $T \sim T'$.

Definition 1.4 (Singly-sorted specification)**.** The specification $S = \langle \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$ is called *singly-sorted* if for any sorts s, s_1, s_2 such that (s, s_1) and (s, s_2) are axioms it follows that $s_1 \equiv s_2$, and for any sorts s_1, s_2, s_3, s_3' such that (s_1, s_2, s_3) and (s_1, s_2, s_3') are rules it follows that $s_3 \equiv s_3'$.

Definition 1.5 (Singly-sorted PTS)**.** A PTS λS is called *singly-sorted* if its specification is singly-sorted.

1.2. **Normalisation in a PTS.** All the properties of PTS*s* above are more or less trivial and can be proved in a generic way. However, there are also non-trivial properties of PTSs which are not generic but depend on the particular PTS that we are considering. The most relevant of them are probably the properties of *weak* and *strong normalisation*.

Definition 1.6 (Normalisation)**.** A PTS is *weakly-normalising* if, starting from any term typable in the PTS considered, it is possible to find a finite β-reduction sequence leading to a normal form. Respectively, the PTS is *strongly-normalising* if all the β-reduction paths starting from any typable term are finite.

From the logical point of view, normalisation is important since it guarantees that the PTS, as a logical system, is consistent whereas, from the computer science point of view, it is important since if a PTS is normalising then the equality between typable terms is decidable.

Strong normalisation is also important since there are cases of PTSs whose only proof of normalisation is in fact a proof of strong normalisation. Moreover, one has to consider that the normalising strategies are not always efficient and some of them can lead to longer paths to the normal forms than others. On the other hand strong normalisation ensures the freedom to choose the preferred reduction strategy since it guarantees that any strategy is normalising.

1.3. **PTSs as logical systems.** In this work we are interested to consider PTSs in the light of their correspondence to logical systems in natural deduction style as introduced by Gentzen in [5].

In order to illustrate such a correspondence let us consider here the natural deduction style system for the minimal fragment of the multi-sorted predicate calculus containing only the implication connective and the universal quantifier[1].

A logical system in natural deduction is usually described by using introduction and elimination rules like the ones in the following table. The corresponding rules in a PTS are obtained by decorating by suitable lambda-terms the rules in natural deduction.

Natural deduction		PTS	
($\to$-int.)	$\dfrac{\Gamma, \alpha \vdash \beta}{\Gamma \vdash \alpha \to \beta}$	(λ-abstr.)	$\dfrac{\Gamma, v{:}\alpha \vdash M : \beta \quad \Gamma \vdash \alpha \to \beta : s}{\Gamma \vdash \lambda v : \alpha.M : \alpha \to \beta}$
($\to$-el.)	$\dfrac{\Gamma \vdash \alpha \to \beta \quad \Gamma \vdash \alpha}{\Gamma \vdash \beta}$	(appl.)	$\dfrac{\Gamma \vdash M : \alpha \to \beta \quad \Gamma \vdash N : \alpha}{\Gamma \vdash MN : \beta}$
($\forall$-int.)	$\dfrac{\Gamma \vdash \beta}{\Gamma \vdash \forall v : \alpha.\beta} \quad v \notin \mathsf{FV}(\Gamma)$	(λ-abstr.)	$\dfrac{\Gamma, v{:}\alpha \vdash M : \beta \quad \Gamma \vdash \Pi v : \alpha.\beta : s}{\Gamma \vdash \lambda v : \alpha.M : \Pi v : \alpha.\beta}$
($\forall$-el.)	$\dfrac{\Gamma \vdash \forall v : \alpha.\beta}{\Gamma \vdash \beta[x := t]}$	(appl.)	$\dfrac{\Gamma \vdash M : \Pi v : \alpha.\beta \quad \Gamma \vdash N : \alpha}{\Gamma \vdash MN : \beta[v := N]}$

Here $\alpha \to \beta$ in the first λ-*abstraction* and *application* in the PTS is an abbreviation for $\Pi v : \alpha.\beta$ in the case the variable v does not appear free in β.

[1]Since a PTS is a typed calculus it is more natural to work in a multi-sorted predicate calculus than in a single sorted one.

If we look at the lambda-terms which are typable in the PTS corresponding to the logical fragment that we are considering it is not immediate to see that they are strongly normalising since a λ-*abstraction* followed by an *application* gives rise to a redex which can not be eliminated in an obvious way.

However there is a logical system which is not using $\rightarrow$-*elimination* and $\forall$-*elimination* and still is equivalent to the one in natural deduction that we considered here, namely, a sequent calculus. In a sequent calculus the rules of introduction and elimination are substituted by rules which introduce the connective or the quantifier on the right or on the left of the $\vdash$ sign, namely, which prescribe how to derive a formula containing such a symbol or how to use an assumption of a formula containing such a symbol.

However, when one wants to decorate the rules of a sequent calculus with lambda-terms in order to obtain a new kind of typing systems, that we will call *Strong Normalising Type Systems* (SNTSs), (s)he has to decide how this should be done in order both to be able to assign a type to all the lambda-terms which are typable in the original system and meanwhile get an easy proof of the strong normalisation property. We propose here the following system which has the main advantage to make immediately clear that the typable lambda-terms are strongly normalising by means of a straightforward proof by induction on the length of the derivation (see theorem 2.4 for a formal proof).

	Seq. Calculus	SNTSs	
($\rightarrow$-left)	$\dfrac{\Gamma \vdash \alpha \quad \Gamma, \beta \vdash \gamma}{\Gamma, \alpha \rightarrow \beta \vdash \gamma}$	$\dfrac{\Gamma \vdash \alpha \rightarrow \beta : s \quad \Gamma \vdash_s t : \alpha \quad \Gamma, y{:}\beta \vdash_s y\vec{p} : \gamma}{\Gamma, x{:}\alpha \rightarrow \beta \vdash_s xt\vec{p} : \gamma}$	
($\rightarrow$-right)	$\dfrac{\Gamma, \alpha \vdash \beta}{\Gamma \vdash \alpha \rightarrow \beta}$	$\dfrac{\Gamma \vdash \alpha \rightarrow \beta : s \quad \Gamma, y{:}\alpha \vdash_s My : \beta}{\Gamma \vdash_s M : \alpha \rightarrow \beta}$	
($\forall$-left)	$\dfrac{\Gamma, \beta[v := t] \vdash \gamma}{\Gamma, \forall v.\beta \vdash \gamma}$	$\dfrac{\Gamma \vdash_s \Pi v : \alpha.\gamma : s \quad \Gamma \vdash_s t : \alpha \quad \Gamma, y{:}\beta[v := t] \vdash_s y\vec{p} : \gamma}{\Gamma, x{:}\Pi v : \alpha.\beta \vdash_s xt\vec{p} : \gamma}$	
($\forall$-right)	$\dfrac{\Gamma \vdash \beta[v := y]}{\Gamma \vdash \forall v.\beta}$	$\dfrac{\Gamma \vdash_s \Pi v : \alpha.\beta : s \quad \Gamma, y{:}\alpha \vdash_s My : \beta[v := y]}{\Gamma \vdash_s M : \Pi v : \alpha.\beta}$	$s \in \mathsf{S}$

However, in order to prove that the new set of rules for the sequent calculus is equivalent to the original one in natural deduction style it is necessary to add to the former the cut rule in order to formalise the idea that natural deduction proofs are closed under composition.

Thus a new rule should be added also to the SNTS system that we are defining, namely

	Seq. Calculus	SNTSs
(cut)	$\dfrac{\Gamma \vdash \alpha \quad \Gamma, \alpha \vdash \beta}{\Gamma \vdash \beta}$	$\dfrac{\Gamma \vdash_s N : \alpha \quad \Gamma, v{:}\alpha \vdash_s M : \beta}{\Gamma \vdash_s M[v := N] : \beta}$

With this addition it is possible to prove the equivalence of the two systems but we lose the easy proof of normalisation of the lambda-terms.

However, it is well known that the story is not finished here since one can prove that the cut-rule is redundant and hence get that the equivalence is in fact between the two systems without the cut-rule. In the case of decorated proofs the cut elimination theorem becomes the proof that the SNTS is closed under substitution and hence one can expect that the strong normalisation theorem for a PTS becomes the proof of closure under substitution of the corresponding SNTS. We will give a complete proof of this result in the next sections.

2. Strong Normalising Type Systems

The previous section suggests that in order to provide a general criterion of strong normalisation for PTSs we could define typing systems in the style of a sequent calculus.

Similar systems has been already used in [4] in order to prove strong normalisation for the simply type lambda calculus, the system F, and a typing system corresponding to a first-order logical calculus. However the generalisation of the previous proofs to a generic PTS has proved to be non-trivial, since the systems introduced in [4] rely on properties of each particular system that cannot be always generalised. Similar systems have been used in [9] in order to address the problem of equivalence between weak and strong normalisation in PTSs.

In the literature there is a certain amount of works dedicated to type systems for sequent calculi. Most of the approaches are based on the introduction of additional constructions with corresponding reductions. For example in [8] and other previous works, typing systems with explicit substitutions are considered. Another example is constituted by the works of Joachimski (see [7]) in which systems with generalised applications are defined.

Our approach in defining *Strong Normalising Type Systems* (SNTSs) is different since we do not introduce additional constructions and reductions keeping the framework as simple as the one of the original PTSs. An approach more similar to ours, but not addressing the relation between cut-elimination and normalisation in a setting with types, is the one presented in [2].

Similarly to PTSs, SNTSs are fully determined by their specifications. In fact, these normalising type systems have the same specifications as PTSs, but different derivation rules which guarantee that they possess the nice property of being strongly-normalising with respect to β-reduction.

As in PTSs the derivable judgements of a SNTS are typing judgements derived under certain assumptions for the free variables, that is, contexts, which have the same properties as the contexts in PTSs. Below, in order to simplify the exposition, we will use the notation $\Gamma + x{:}T$ to denote the following context.

$$\Gamma + x{:}T = \begin{cases} \Gamma & \text{if } x{:}T \in \Gamma; \\ \Gamma, x{:}T & \text{if } x \notin FV(\Gamma); \\ \text{undefined} & \text{if } x{:}U \in \Gamma, \text{ and } U \not\equiv T \end{cases}$$

The intended meaning of the above notation is denoting a context in which $x{:}T$ already appears or is added in the end of the list of pairs.

Definition 2.1. A *Strong Normalising Type System* is a pair $\lambda^s S = \langle S, \vdash_s \rangle$, where $S = \langle \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$ is a specification and $\vdash_s$ is the judgement relation of $\lambda^s S$ determined by the following derivation rules:

(axioms)	$\vdash_s s_1 : s_2$	$(s_1, s_2) \in \mathsf{A}$
(Π-formation)	$\dfrac{\Gamma \vdash_s T : s_1 \quad \Gamma, v{:}T \vdash_s U : s_2}{\Gamma \vdash_s \Pi v : T.U : s_3}$	$(s_1, s_2, s_3) \in \mathsf{R}$
(variables)	$\dfrac{\Gamma \vdash_s T : s}{\Gamma, v{:}T \vdash_s v : T}$	$s \in \mathsf{S}, v \notin FV(\Gamma)$
(Π-left)	$\dfrac{\Gamma \vdash_s \Pi v : T_1.T_2 : s \quad \Gamma \vdash_s M : T_1 \quad \Gamma, y{:}T_2[v := M] \vdash_s y\vec{p} : U}{\Gamma + x{:}\Pi v : T_1.T_2 \vdash_s xM\vec{p} : U}$	$y \notin FV(\Gamma, \vec{p}, U)$
(Π-right)	$\dfrac{\Gamma \vdash_s \Pi v : T.U : s \quad \Gamma, v{:}T \vdash_s Mv : U}{\Gamma \vdash_s M : \Pi v : T.U}$	$s \in \mathsf{S}, v \notin FV(\Gamma, M)$
(λ-expansion)	$\dfrac{\Gamma \vdash_s M[x := N]\vec{p} : U \quad \Gamma \vdash_s N : T \quad \Gamma, x{:}T \vdash_s M : V \quad \Gamma \vdash_s \Pi x : T.V : s}{\Gamma \vdash_s (\lambda x : T.M)N\vec{p} : U}$	$s \in \mathsf{S}$
(type expansion)	$\dfrac{\Gamma \vdash_s M : T \quad \Gamma \vdash_s U : s}{\Gamma \vdash_s M : U}$	$U \twoheadrightarrow T$
(context expansion)	$\dfrac{\Gamma, x{:}T, \Gamma' \vdash_s M : U \quad \Gamma \vdash_s T' : s}{\Gamma, x{:}T', \Gamma' \vdash_s M : U}$	$T' \twoheadrightarrow T$
(weakening)	$\dfrac{\Gamma \vdash_s T : s \quad \Gamma \vdash_s M : U}{\Gamma, v{:}T \vdash_s M : U}$	$s \in \mathsf{S}, v \notin FV(\Gamma)$
(permutation)	$\dfrac{\Gamma, x{:}T_1, y{:}T_2, \Gamma' \vdash_s M : U}{\Gamma, y{:}T_2, x{:}T_1, \Gamma' \vdash_s M : U}$	$x \notin FV(T_2)$

It is worth noting immediately that with respect to the discussion of the previous section which leaded us to formulate the rules for a SNTS, in the formal definition above a new rule appeared, namely, λ-*expansion*, which has no logical counterpart. Indeed, it is there to allow us to introduce redexes in our typing systems, which is necessary in order to show that SNTSs are equivalent to strongly normalising PTSs with the same specification. However, as we will see in the strong normalisation theorem for SNTSs, the redexes introduced by a λ-*expansion* are totally safe and they will never lead to a non normalising lambda-term.

Note also that the rule for *conversion* of a PTS is here substituted by a rule for *expansion* which seems to be weaker. In the next section we will prove that also the *reduction* rule is valid, and thus it follows that the *conversion* rule is admissible for SNTSs.

As we promised, it is now easy to show that any SNTS enjoys strong normalisation. Indeed a trivial analysis of the rules show that the following lemmas hold.

Lemma 2.2 (Correctness of types)**.** Let $\lambda^s S$ be a SNTS and let $\Gamma \vdash_s M : T$. Then $T \in \mathsf{S}$ or there exists a sort s, such that the judgement $\Gamma \vdash_s T : s$ appears up in the derivation of $\Gamma \vdash_s M : T$.

Lemma 2.3 (Inversion Lemma for Types)**.** Let $\lambda^s S$ be a SNTS. Then, if $\Gamma \vdash_s \Pi v : T_1.T_2 : s$ is a derivable judgement then also $\Gamma \vdash_s T : s_1$ and $\Gamma, v{:}T_1 \vdash_s T_2 : s_2$ are derivable for some rule $(s_1, s_2, s) \in \mathsf{R}$.

Hence the strong normalisation theorem immediately follows.

Theorem 2.4 (Normalisation)**.** Any SNTS is strongly-normalising.

Proof: The proof of this theorem is straightforward by induction on the length of derivations in a SNTS. The only non-trivial case is the one of the λ-*expansion* rule. Indeed, in order to prove that $(\lambda x : T.M)N\vec{p}$ is strongly normalising, we need to know that T, $M[x := N]\vec{p}$ and N are strongly normalising (it is easy to see that a redex in $(\lambda x : T.M)N\vec{p}$ gives rise to a redex in T, $M[x := N]\vec{p}$ or N). Strong normalisation for the last two terms follows directly from the induction hypothesis. Moreover, since the judgement $\Gamma \vdash_s N : T$ is derivable, from Lemma 2.2 it follows that $T \equiv s$, for some sort s, and thus T is strongly normalising or that there exists a derivation of the judgement $\Gamma \vdash_s T : s$ shorter than the one of the judgement $\Gamma \vdash_s N : T$ and thus, applying the induction hypothesis to the derivation of this judgement, it follows that T is strongly normalising. ■

2.1. **Basic Properties of SNTSs.** In this section we will prove those properties of SNTSs that are necessary in order to show our main result, namely, theorem 2.8. The most important among these properties is next lemma of closure of the typed judgements under reduction at the level of contexts, terms and types.

Lemma 2.5 (Reduction Lemma for SNTSs)**.** Let $\lambda^s S$ be a SNTS with specification $S \equiv \langle \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$. Then, if $\Gamma \vdash_s M : T$ and $\Gamma \twoheadrightarrow \Gamma'$, $M \twoheadrightarrow M'$ and $T \twoheadrightarrow T'$, then there exist a derivation of the judgement $\Gamma' \vdash_s M' : T'$ which is no longer than the one of $\Gamma \vdash_s M : T$.

Proof: The proof is by induction on the length of the derivation of $\Gamma \vdash_s M : T$. We will use the notation $\Gamma \vdash_s^n M : T$ to mean that the derivation length of the judgement $\Gamma \vdash_s M : T$ is less or equal to n (note that if $\Gamma \vdash_s^n M : T$, then $\Gamma \vdash_s^{n+1} M : T$). Below the non-trivial induction steps are considered in detail. In the rest of the induction cases we will consider the minimum n such that $\Gamma \vdash_s^{n+1} M : T$ holds.

Let us start by considering the Π-left rule:

$$(\Pi\text{-left})\ \frac{\Gamma \vdash_s^n \Pi v : T_1.T_2 : s \quad \Gamma \vdash_s^n M : T_1 \quad \Gamma, y{:}T_2[v := M] \vdash_s^n y\vec{p} : U}{\Gamma + x{:}\Pi v : T_1.T_2 \vdash_s^{n+1} xM\vec{p} : U},$$

where $y \notin FV(\Gamma, \vec{p}, U)$. Let now $\Gamma + x{:}\Pi v : T_1.T_2 \twoheadrightarrow \Gamma' + x{:}\Pi v : T_1'.T_2'$ and $xM\vec{p} \twoheadrightarrow xM'\vec{p}'$ and $U \twoheadrightarrow U'$. In this case $\Gamma \twoheadrightarrow \Gamma'$ and $\Pi v : T_1.T_2 \twoheadrightarrow \Pi v : T_1'.T_2'$, thus by the induction hypothesis for $\Gamma \vdash_s^n \Pi v : T_1.T_2 : s$ it follows

$$\Gamma' \vdash_s^n \Pi v : T_1'.T_2' : s \tag{2.1}$$

Moreover, since $\Gamma \twoheadrightarrow \Gamma'$, $M \twoheadrightarrow M'$, $T_1 \twoheadrightarrow T_1'$, then applying the induction hypothesis to $\Gamma \vdash_s^n M : T_1$ we obtain:

$$\Gamma' \vdash_s^n M' : T_1' \tag{2.2}$$

In the end, from $T_2 \twoheadrightarrow T_2'$ and $M \twoheadrightarrow M'$ it follows $T_2[v := M] \twoheadrightarrow T_2'[v := M']$, and hence $\Gamma, y{:}T_2[v := M] \twoheadrightarrow \Gamma', y{:}T_2'[v := M']$. Since also $y\vec{p} \twoheadrightarrow y\vec{p}'$ and $U \twoheadrightarrow U'$ applying the induction hypothesis to $\Gamma, y{:}T_2[v := M] \vdash_s^n y\vec{p} : U$ we obtain

$$\Gamma', y{:}T_2'[v := M'] \vdash_s^n y\vec{p}' : U' \tag{2.3}$$

Then, from 2.1, 2.2 and 2.3 it follows, $\Gamma', x{:}\Pi v : T_1'.T_2' \vdash_s^{n+1} xM'\vec{p}' : U'$.

Let us consider now the λ-expansion rule:

$$(\lambda\text{-expansion})\ \frac{\Gamma \vdash_s^n M[x := N]\vec{p} : U \quad \Gamma \vdash_s^n N : T \quad \Gamma, x{:}T \vdash_s^n M : V \quad \Gamma \vdash_s^n \Pi x : T.V : s}{\Gamma \vdash_s^{n+1} (\lambda x : T.M)N\vec{p} : U}$$

Let $\Gamma \twoheadrightarrow \Gamma'$, $U \twoheadrightarrow U'$ and $(\lambda x : T.M)N\vec{p} \twoheadrightarrow R$. There are two possibilities for the last reduction: $(\lambda x : T.M)N\vec{p} \twoheadrightarrow (\lambda x : T'.M')N'\vec{p}' \equiv R$ or $(\lambda x : T.M)N\vec{p} \twoheadrightarrow (\lambda x : T'.M')N'\vec{p}' \rightsquigarrow M'[x := N']\vec{p}' \twoheadrightarrow R$

To begin with let us assume that $(\lambda x : T.M)N\vec{p} \twoheadrightarrow (\lambda x : T'.M')N'\vec{p}' \equiv R$. In this case we have that $M \twoheadrightarrow M'$, $N \twoheadrightarrow N'$, $\vec{p} \twoheadrightarrow \vec{p}'$ and hence

$$M[x := N]\vec{p} \twoheadrightarrow M'[x := N']\vec{p}'.$$

Since also $U \twoheadrightarrow U'$, $T \twoheadrightarrow T'$, $\Gamma \twoheadrightarrow \Gamma'$, and $\Pi x : T.V \twoheadrightarrow \Pi x : T'.V$ from the induction hypothesis applied to the judgements in the premises of the λ-expansion rule, namely, $\Gamma \vdash_s^n M[v := N]\vec{p} : U$ and $\Gamma \vdash_s^n N : T$ and $\Gamma, x{:}T \vdash_s^n M : V$ and $\Gamma \vdash_s^n \Pi x : T.V : s$ it follows

$$\Gamma' \vdash_s^n M'[v := N']\vec{p}' : U',\ \Gamma' \vdash_s^n N' : T',\ \Gamma', x{:}T' \vdash_s^n M' : V \text{ and } \Gamma' \vdash_s^n \Pi x : T'.V : s$$

and hence, by an application of λ-expansion, we get $\Gamma' \vdash_s^{n+1} (\lambda x : T'.M')N'\vec{p}' : U'$, that is $\Gamma' \vdash_s^{n+1} R : U'$.

Let now $(\lambda x : T.M)N\vec{p} \twoheadrightarrow (\lambda x : T'.M')N'\vec{p}' \leadsto M'[x := N']\vec{p}' \twoheadrightarrow R$. In this case we have that $M[x := N]\vec{p} \twoheadrightarrow M'[x := N']\vec{p}' \twoheadrightarrow R$ and thus by the induction hypothesis for $\Gamma \vdash_s^n M[x := N]\vec{p} : U$ we obtain $\Gamma' \vdash_s^n R : U'$ and hence $\Gamma' \vdash_s^{n+1} R : U'$.

The last rule we show is the type-espansion rule:

$$\text{(type expansion)}\ \frac{\Gamma \vdash_s^n M : T \quad \Gamma \vdash_s^n U : s}{\Gamma \vdash_s^{n+1} M : U} \qquad U \twoheadrightarrow T$$

Let $\Gamma \twoheadrightarrow \Gamma'$, $M \twoheadrightarrow M'$ and $U \twoheadrightarrow U'$. Since $U \twoheadrightarrow T$, from the Church-Rosser property of β-reduction it follows that there exists T' such that $T \twoheadrightarrow T'$ and $U' \twoheadrightarrow T'$. Then from the induction hypothesis applied to the judgement $\Gamma \vdash_s^n M : T$ it follows $\Gamma' \vdash_s^n M' : T'$ and from the induction hypothesis applied to the judgement $\Gamma \vdash_s^n U : s$ it follows $\Gamma' \vdash_s^n U' : s$. Since $U' \twoheadrightarrow T'$ applying the type-expansion rule we obtain $\Gamma' \vdash_s^{n+1} M' : U'$. ■

The Reduction Lemma 2.5 shows that the rules for type and context expansion are sufficient in order to get that the corresponding conversion rules are admissible for the SNTSs.

Corollary 2.6 (Admissibility of the conversion rule)**.** If $\Gamma \vdash_s M : T$, $\Gamma \vdash_s U : s$ and $T \sim U$, then $\Gamma \vdash_s M : U$.

Proof: From the confluence property for the $\sim$ relation it follows that there exists a term T' such that $T \twoheadrightarrow T'$ and $U \twoheadrightarrow T'$. Then, from the Reduction Lemma it follows $\Gamma \vdash_s M : T'$ and thus by applying the rule for type expansion we obtain $\Gamma \vdash_s M : U$. ■

2.2. **A Strong Normalisation Condition for PTSs.** Finally we can prove our main theorem. In order to state it let us introduce the following definition.

Definition 2.7. A SNTS $\lambda^s S$ is *closed under substitution* if $\Gamma \vdash_s M[x := N] : U[x := N]$ holds whenever both $\Gamma, x{:}T \vdash_s M : U$ and $\Gamma \vdash_s N : T$ hold.

Then we can prove the following theorem.

Theorem 2.8 (Strong normalisation condition)**.** If the SNTS $\lambda^s S$ is closed under substitution, then the corresponding PTS λS is strongly normalising.

In order to prove this result, we will make use of the following lemma.

Lemma 2.9. Let $\lambda^s S$ be a SNTS. Then, if $\Gamma \vdash_s M : \Pi v : T_1.T_2$ then $\Gamma, v{:}T_1 \vdash_s Mv : T_2$, where $v \notin FV(\Gamma)$.

Proof: The proof is by induction on the length of the derivation of $\Gamma \vdash_s M : \Pi v : T_1.T_2$. The only non trivial cases are when the last rule applied in this derivation is the *variable* or the Π-*left* rule. Let us consider first the *variable* rule, that is,

$$\frac{\Gamma \vdash_s \Pi v : T_1.T_2 : s}{\Gamma, x{:}\Pi v : T_1.T_2 \vdash_s x : \Pi v : T_1.T_2} \quad s \in \mathsf{S},\ x \notin \mathsf{FV}(\Gamma)$$

We want to prove that the judgement $\Gamma, x{:}\Pi v : T_1.T_2, v{:}T_1 \vdash_s xv : T_2$ is derivable, for $v \notin FV(\Gamma) \bigcup FV(\Pi v : T_1.T_2)$. Note that since $v \notin FV(\Pi v : T_1.T_2)$ it will be sufficient to prove that the judgement $\Gamma, v{:}T_1, x{:}\Pi v : T_1.T_2 \vdash_s xv : T_2$ is derivable. Now, by Lemma 2.3, from $\Gamma \vdash_s \Pi v : T_1.T_2 : s$ it follows that $\Gamma \vdash_s T_1 : s_1$ for some sort s_1 and hence

$$\Gamma, v{:}T_1 \vdash_s v : T_1, \tag{2.4}$$

where v is the variable considered above. Moreover, again by Lemma 2.3, $\Gamma \vdash_s \Pi v : T_1.T_2 : s$ yields also $\Gamma, v{:}T_1 \vdash_s T_2 : s_2$, and hence applying the *variable* rule we obtain

$$\Gamma, v{:}T_1, y{:}T_2 \vdash_s y : T_2, \tag{2.5}$$

Thus, applying the Π-*left* rule to the above two judgements we obtain

$$\Gamma, v{:}T_1, x{:}\Pi v : T_1.T_2 \vdash_s xv : T_2$$

Let us consider now the case when the last rule in the derivation under consideration is the Π-left rule, that is,

$$\frac{\Gamma \vdash_s \Pi v : T_1.T_2 : s \quad \Gamma \vdash_s M : T_1 \quad \Gamma, y{:}T_2[v := M] \vdash_s y\vec{p} : \Pi u : U_1.U_2}{\Gamma + x{:}\Pi v : T_1.T_2 \vdash_s xM\vec{p} : \Pi u : U_1.U_2},$$

where $y \notin FV(\Gamma) \bigcup FV(\vec{p}) \bigcup (\Pi u : U_1.U_2)$. From the induction hypothesis it follows

$$\Gamma, y{:}T_2[v := M], u{:}U_1 \vdash_s y\vec{p}u : U_2, \tag{2.6}$$

and since $y \notin FV(U_1)$ by applying the *permutation* rule it follows

$$\Gamma, u{:}U_1, y{:}T_2[v := M] \vdash_s y\vec{p}u : U_2, \tag{2.7}$$

Therefore, by applying the Π-*left* rule it follows

$$\Gamma, u{:}U_1 + x{:}\Pi v : T_1.T_2 \vdash_s xM\vec{p}u_1 : \Pi u : U_1.U_2 \ \blacksquare$$

Now we can proceed with the proof of theorem 2.8. We will show that if the SNTS $\lambda^s S$ is closed under substitution then any judgement derivable in the corresponding PTS λS is also derivable in $\lambda^s S$; it is clear that this is sufficient since we already proved that all the terms typable in a SNTS are strongly normalisable.

The proof will be by induction on the length of the derivation of the judgement in the PTS. The cases for *axioms*, *variable* rules, Π-*formation* and *weakening* are trivial. The case of the *conversion* rule follows directly from Corollary 2.6. The remaining cases, namely, the *application* and *abstraction* rules, are considered here below.

So, suppose that the last rule in the derivation under consideration is an instance of the *application* rule, that is,

$$\frac{\Gamma \vdash M : \Pi v : T_1.T_2 \quad \Gamma \vdash N : T_1}{\Gamma \vdash MN : T_2[v := N]}$$

By induction hypothesis it follows that $\Gamma \vdash_s M : \Pi v : T_1.T_2$ and $\Gamma \vdash_s N : T_1$. Consequently $\Gamma, v{:}T_1 \vdash_s Mv : T_2$ follows by lemma 2.9. Now, by assumption $\lambda^s S$ is closed under substitution, and thus we obtain $\Gamma \vdash_s MN : T_2[v := N]$.

Finally, let us consider the case of the *abstraction* rule, that is,

$$\frac{\Gamma, v{:}T \vdash M : U \quad \Gamma \vdash \Pi v : T.U : s}{\Gamma \vdash \lambda v : T.M : \Pi v : T.U}$$

By induction hypothesis it follows both that $\Gamma, v{:}T \vdash_s M : U$ and $\Gamma \vdash_s \Pi v : T.U : s$. Then also $\Gamma \vdash_s T : s'$, for some sort s', and hence $\Gamma, v{:}T \vdash_s v : T$, by applying the *variable* rule. Thus we can apply the λ-*expansion* rule to the judgements $\Gamma, v{:}T \vdash_s M : U$, $\Gamma, v{:}T \vdash_s v : T$, $\Gamma, v{:}T \vdash_s M : U$ and $\Gamma \vdash_s \Pi v : T.U : s$ obtaining $\Gamma, v{:}T \vdash_s (\lambda v : T.M)v : U$ and hence we can conclude $\Gamma \vdash_s \lambda v : T.M : \Pi v : T.U$ by an instance of the Π-*right* rule.

Thus the proof of theorem 2.8 is completed.

3. From a condition to a criterion

In order to transform the condition for strong normalisability for a PTS that we proposed in the previous sections into a real criterion one should show that if a PTS is strongly normalising then the corresponding SNTS is indeed closed under substitution.

The easiest way to get this result is to show that the rules for an SNTS are sufficient to get all the judgements provable in the corresponding PTS but using the assumption of strong normalisability of the PTS instead of the one of closure under substitution of the SNTS (this approach was used in [11] in order to show that a calculus for the intersection types similar to the one that we are proposing here can type all the strongly normalising lambda terms). Indeed, it is not difficult to prove the following lemma by induction on the length of the proof of the considered judgement.

Lemma 3.1. Let $\Lambda^s S$ be a SNTS. Then if $\Gamma \vdash_s N : T$ holds then $\Gamma \vdash N : T$ is provable in the corresponding PTS.

Hence if $\Gamma, v{:}T \vdash_s M : U$ and $\Gamma \vdash_s N : T$ then $\Gamma, v{:}T \vdash M : U$ and $\Gamma \vdash N : T$ are provable in the corresponding PTS; thus $\Gamma \vdash M[v := N] : U[v := N]$ follows, since any PTS is closed under substitution, and so $\Gamma \vdash_s M[v := N] : U[v := N]$ would follow as a consequence of the sufficiency of the rules of the SNTS.

Observe now that most of the proofs of closure of the SNTS for the rules of the corresponding PTS can be obtained with no reference to closure under substitution and that the only step which uses such an assumption is proving closure under the *application* rule. Thus, one only needs to prove such a closure by using the property of strong normalisability of the corresponding PTS. The proof of this statement is still an open problem even if we guess that such a result can be obtained.

In this way our condition would become a complete criterion and hence the failure of being closed under substitution for a SNTS could be used in order to show that the corresponding PTS is not strongly normalising.

References

[1] H. Barendregt. Typed lambda calculi. In Abramski, editor, *Handbook of Logic in Computer Science*. Oxford University Press, 1992.

[2] H. Barendregt and S. Ghilezan. Lambda terms for natural deduction, sequent calculus and cut elimination. *Functional Programming*, 80:121–134, 2000.

[3] S. Berardi. *Type dependence and constructive mathematics*. PhD thesis, Dipartamento Matematica, Universita di Torino, 1990.

[4] V. Capretta and S. Valentini. A general method to prove normalisation theorem for first and second order typed λ-calculi. *Mathematical Structures in Computer Science*, 9:719–739, 1999.

[5] G. Gentzen. Untersuchungen über das logischen schliessen. In M. E. Szabo, editor, *Collected papers of Gerhard Gentzen, translations*. North-Holland, Amsterdam, 1969, 1935.

[6] H. Geuvers. *Logics and Type Systems*. PhD thesis, Katholieke Universiteit Nijmegen, The Netherlands, 1993.

[7] F. Joachimski. On Zucher's isomorphism for LJ and its extension to pure type systems. *Submitted to Journal of Functional Programming*, 2003.

[8] R. Kervarc. Pure type systems and classical sequents. Technical report, Ecole Normale Superieure de Lyon, 2006.

[9] M. T. Stefanova. *Properties of Typing Systems*. PhD thesis, University of Nijmegen, 1999.

[10] J. Terlouw. Een nadere bewijstheoretische analyse van GSTT's. Technical report, University of Nijmegen, 1989.

[11] S. Valentini. An elementary proof of strong normalization for intersection types. *Archive for Mathematical Logic*, 40:475–488, 2001.