2001

Pure type systems with more liberal rules

Martin W. Bunder
University of Wollongong, mbunder@uow.edu.au

Wil Dekkers
Catholic University

Publication Details

Bunder, M. & Dekkers, W. (2001). Pure type systems with more liberal rules. Journal of Symbolic Logic, 66 (4), 1561-1580.

Pure type systems with more liberal rules

Abstract
Pure Type Systems. PTSs, introduced as a generalisation of the type systems of Barendregt's lambda-cube, provide a foundation for actual proof assistants, aiming at the mechanic verification of formal proofs. In this paper we consider simplifications of some of the rules of PTSs. This is of independent interest for PTSs as this produces more flexible PTS-like systems, but it will also help, in a later paper, to bridge the gap between PTSs and systems of Illative Combinatory Logic.

First we consider a simplification of the start and weakening rules of PTSs. which allows contexts to be sets of statements, and a generalisation of the conversion rule. The resulting Set-modified PTSs or SPTSs, though essentially equivalent to PTSs, are closer to standard logical systems.

A simplification of the abstraction rule results in Abstraction-modified PTSs or APTSs. These turn out to be equivalent to standard PTSs if and only if a condition (*) holds. Finally we consider SAPTSs which have both modifications.

Keywords
liberal, rules, more, pure, systems, type

Disciplines
Engineering | Science and Technology Studies

Publication Details
Bunder, M. & Dekkers, W. (2001). Pure type systems with more liberal rules. Journal of Symbolic Logic, 66 (4), 1561-1580.

THE JOURNAL OF SYMBOLIC LOGIC
Volume 66, Number 4, Dec. 2001

PURE TYPE SYSTEMS WITH MORE LIBERAL RULES

MARTIN BUNDER AND WIL DEKKERS

Abstract. Pure Type Systems, PTSs, introduced as a generalisation of the type systems of Barendregt's lambda-cube, provide a foundation for actual proof assistants, aiming at the mechanic verification of formal proofs. In this paper we consider simplifications of some of the rules of PTSs. This is of independent interest for PTSs as this produces more flexible PTS-like systems, but it will also help, in a later paper, to bridge the gap between PTSs and systems of Illative Combinatory Logic.

First we consider a simplification of the start and weakening rules of PTSs, which allows contexts to be sets of statements, and a generalisation of the conversion rule. The resulting Set-modified PTSs or SPTSs, though essentially equivalent to PTSs, are closer to standard logical systems.

A simplification of the abstraction rule results in Abstraction-modified PTSs or APTSs. These turn out to be equivalent to standard PTSs if and only if a condition $(*)$ holds. Finally we consider SAPTSs which have both modifications.

§1. Introduction. Pure Type Systems, PTSs, introduced by Berardi [5] and Terlouw [25] as a generalisation of the type systems of Barendregt's λ-cube (cf. Barendregt [2]), provide a foundation for proof checking systems like AUTOMATH, Coq and Lego. PTSs have a common framework of postulates and each individual PTS is determined by its *specification*, which consists of a set $\mathcal{S}$ of sorts, a set $\mathcal{A}$ of axioms and a set $\mathcal{R}$ of triples of sorts over which the product rule holds. (For details see Barendregt [2] or Section 2 below.)

In Sections 2 and 3 we give a short introduction to PTSs. We quote some lemmas from Barendregt [2] and add some useful new lemmas. In Section 4 we consider a simplification of the axioms and start-rule of PTSs, which allows contexts to be *sets* of statements, and a generalisation of the conversion rule. We show in Section 5 that these Set-modified PTSs or SPTSs are in a strong sense equivalent to standard PTSs with the same specification.

A simplification of the abstraction rule, in Section 6, results in Abstraction-modified PTSs or APTSs. In Sections 7 and 8 these APTSs turn out to be equivalent to standard PTSs with the same specification if a condition $(*)$ on the set of rules $\mathcal{R}$ holds. Moreover we show that for a singly sorted PTS and an APTS with the same specification $(*)$ is not only a sufficient but also a necessary condition for the equivalence to hold.

Systems for which $(*)$ holds include $\lambda\!\rightarrow$, that represents Church's original type theory of 1940 (see Church [10]), $\lambda 2$, the second order typed lambda calculus of

Received July7, 1999.
Research supported by the Australian Research Council.

0022-4812/01/6604-0004/$3.00

Girard [16], λC, the Calculus of Constructions of Coquand and Huet [11] as well as the de Bruijn AUTOMATH PTSs, λP, λ-AUT-68, λ-AUT-QE and λPAL (see de Bruijn [12] for a survey) and in a certain sense the system ECC of Luo [19].

Then in Section 9 we consider SAPTSs which have both modifications. Of course these again are equivalent to PTSs with the same specification provided $(*)$ holds.

We summarize the results on the equivalencies between PTSs, SPTSs, APTSs and SAPTSs in Section 10.

The work on APTSs generalises the work of Pollack [22], van Benthem Jutting, McKinna and Pollack [4], Severi [24] and Poll [21] on PTSs, which was motivated by the problem of finding reasonable algorithms for type checking Pure Type Systems that are in use, and to implement (efficient) sound and complete type checkers. Our aim is to investigate if the derivation rules for PTSs can be replaced by more liberal rules in such a way that the resulting systems differ only slightly, if at all, from the original PTSs. This leads to more flexible PTS-like systems, which are closer to their formulas as types interpretations. Moreover we will show in a later paper that each SAPTS is equivalent, in a strong sense, to a form of Illative Combinatory Logic. (See Barendregt, Bunder and Dekkers [3] for an introduction.)

§2. Pure type systems. These are formal systems which allow the derivation of *judgements* of the form:

$$\Gamma \vdash M : A$$

where M and A are *pseudoterms*, $M : A$ a *statement*, interpreted as M has the type A and Γ a sequence of certain statements called a *context*. More formally:

DEFINITION 2.1. The class of *pseudoterms* $\mathscr{T}$ is given by:

$$\mathscr{T} = \mathrm{V} \mid \mathscr{C} \mid (\Pi \mathrm{V} : \mathscr{T} . \mathscr{T}) \mid (\lambda \mathrm{V} : \mathscr{T} . \mathscr{T}) \mid \mathscr{T}\mathscr{T}$$

where $\mathrm{V} = \{x, y, z, \ldots, x_1, x_2, \ldots\}$ is a class of variables and $\mathscr{C} = \{c, c_1, c_2, \ldots\}$ is a class of constants.

If $x \in \mathrm{V}$ and $t_1, t_2 \in \mathscr{T}$, $(\lambda x : t_1 . t_2)$ is interpreted as the λ-abstraction of t_2 with respect to the variable x of type t_1 and $(\Pi x : t_1 . t_2)$ is interpreted as the class (or type) of all generalised functions from t_1 to t_2, where t_2 may be dependent on the argument x of the function. In $(\Pi x : t_1 . t_2)$ x is bound just as in $(\lambda x : t_1 . t_2)$. $\mathrm{FV}(t)$ will denote the set of free variables of t.

DEFINITION 2.2.

(i) If M and A are pseudoterms $M : A$ is a *statement*.

(ii) Γ is a *context* if it is a sequence of statements $\langle x_1 : A_1, \ldots, x_n : A_n \rangle$ where $x_1, \ldots, x_n \in \mathrm{V}$. We will let $\mathrm{FV}(\Gamma)$ be the set of free variables of the pseudoterms in Γ.

(iii) If Γ is a context and M and A are pseudoterms then $\Gamma \vdash M : A$ is a *judgement*.

DEFINITION 2.3 (Pure Type Systems (PTSs)).

(i) The *specification* of a PTS consists of a triple $\mathrm{S} = (\mathscr{S}, \mathscr{A}, \mathscr{R})$ where $\mathscr{S}$ is a subclass of $\mathscr{C}$, called the *sorts*, $\mathscr{A}$ is a class of statements of the form $(c : s)$ and $\mathscr{R}$ is a subclass of $\mathscr{S} \times \mathscr{S} \times \mathscr{S}$.

(ii) A *Pure Type System* (PTS) $\lambda S = \lambda(\mathcal{S}, \mathcal{A}, \mathcal{R})$ determined by the specification $S = (\mathcal{S}, \mathcal{A}, \mathcal{R})$ is defined as follows. Statements and contexts are as in Definition 2.2. The notion of type derivation, written as $\Gamma \vdash^{\lambda S} M : A$ (or just $\Gamma \vdash M : A$), is defined by the following postulates:

(axioms) $$\langle\,\rangle \vdash c : s \qquad (c : s) \in \mathcal{A};$$

(start rule) $$\frac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A} \qquad x \notin \mathrm{FV}(\Gamma);$$

(weakening rule) $$\frac{\Gamma \vdash M : A \quad \Gamma \vdash B : s}{\Gamma, x : B \vdash M : A} \qquad x \notin \mathrm{FV}(\Gamma);$$

(product rule) $$\frac{\Gamma \vdash A : s_1 \quad \Gamma, x : A \vdash B : s_2}{\Gamma \vdash (\Pi x : A.B) : s_3} \qquad (s_1, s_2, s_3) \in \mathcal{R};$$

(abstraction rule) $$\frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash (\Pi x : A.B) : s}{\Gamma \vdash (\lambda x : A.M) : (\Pi x : A.B)} \qquad s \in \mathcal{S};$$

(application rule) $$\frac{\Gamma \vdash M : (\Pi x : A.B) \quad \Gamma \vdash N : A}{\Gamma \vdash (MN) : B[x := N]}\ ;$$

(conversion rule) $$\frac{\Gamma \vdash M : A \quad \Gamma \vdash B : s \quad A =_\beta B}{\Gamma \vdash M : B} \qquad s \in \mathcal{S}.$$

Any $s, s_1, s_2, \ldots$ used below will be assumed to be an element of $\mathcal{S}$.

DEFINITION 2.4. A context Γ is said to be *legal* in a PTS λS if there are pseudoterms M and A such that in λS: $\Gamma \vdash M : A$.

We assume knowledge of basic λ-calculus (for details see Barendregt [1] or Hindley and Seldin [17]). We extend the notions of reduction and subsets to contexts as follows.

DEFINITION 2.5. If $\Gamma \equiv \langle x_1 : A_1, \ldots, x_n : A_n\rangle$ and $\Gamma_1 \equiv \langle x_1 : B_1, \ldots, x_n : B_n\rangle$ are contexts then $\Gamma \twoheadrightarrow_\beta \Gamma_1$ $(\Gamma =_\beta \Gamma_1)$ if for $1 \le i \le n$ $A_i \twoheadrightarrow_\beta B_i$ $(A_i =_\beta B_i)$.

DEFINITION 2.6. Γ is *part* of Γ_1 $(\Gamma \subseteq \Gamma_1)$ if every $x : A$ in Γ is also in Γ_1.

The statement on terms in the following theorem is well known and it can easily be generalised to contexts.

THEOREM 2.7 (The Church-Rosser Theorem for Pseudoterms and Contexts).

(i) *If* $M_1 =_\beta M_2$ *then there is an* M_3 *such that* $M_1 \twoheadrightarrow_\beta M_3$ *and* $M_2 \twoheadrightarrow_\beta M_3$.
(ii) *If* $\Gamma_1 =_\beta \Gamma_2$ *then there is a* Γ_3 *such that* $\Gamma_1 \twoheadrightarrow_\beta \Gamma_3$ *and* $\Gamma_2 \twoheadrightarrow_\beta \Gamma_3$.

§3. Some properties of PTSs. Lemmas 3.1, 3.2, 3.3, 3.5, 3.8 and 3.11 are taken from Barendregt [2]. The other lemmas are new as far as we know.

LEMMA 3.1 (Start Lemma). *If* Γ *is a legal context then*

(i) $(c : s) \in \mathcal{A} \Longrightarrow \Gamma \vdash c : s$;
(ii) $(x : A) \in \Gamma \Longrightarrow \Gamma \vdash x : A$.

LEMMA 3.2 (Substitution Lemma).

$$\Gamma, x: A \vdash B: C \ \&\ \Gamma \vdash D: A \Longrightarrow \Gamma \vdash B[x := D]: C[x := D].$$

LEMMA 3.3 (Subject Reduction Theorem).

$$\Gamma \vdash M: A,\ \Gamma \twoheadrightarrow_\beta \Gamma',\ M \twoheadrightarrow_\beta M',\ A \twoheadrightarrow_\beta A' \Longrightarrow \Gamma' \vdash M': A'.$$

The following lemma can be useful in proofs by induction on (the length of) a derivation like in the proof of the Thinning Lemma 3.5.

LEMMA 3.4. *If* $x_1: A_1, \ldots, x_n: A_n \vdash M: A$ *then for each* i, $1 \leq i \leq n$, *there is an* $s_i \in \mathcal{S}$ *such that the derivation of* $x_1: A_1, \ldots, x_n: A_n \vdash M: A$ *contains a derivation of* $x_1: A_1, \ldots, x_{i-1}: A_{i-1} \vdash A_i: s_i$.

PROOF. By induction on the derivation of $x_1: A_1, \ldots, x_n: A_n \vdash M: A$. ⊣

LEMMA 3.5 (Thinning Lemma). *If* Γ *and* Γ' *are legal contexts and* $\Gamma \subseteq \Gamma'$ *then*

$$\Gamma \vdash M: A \Longrightarrow \Gamma' \vdash M: A.$$

REMARK. We are not aware of a correct proof in the literature of the Thinning Lemma. The proof is by induction on the (length of the) derivation of $\Gamma \vdash M: A$. In the case of the abstraction rule

$$\Gamma \vdash M: A \quad \text{is} \quad \Gamma \vdash \lambda x: C.N: \Pi x: C.D$$

as consequence of

$$\Gamma, x: C \vdash N: D, \quad \Gamma \vdash \Pi x: C.D: s$$

we need that $\Gamma', x: C$ is legal. This we get as follows: By our Lemma 3.4 the derivation of $\Gamma, x: C \vdash N: D$ contains a derivation of $\Gamma \vdash C: s_1$ for some s_1. Hence we get by the induction hypothesis $\Gamma' \vdash C: s_1$ and hence $\Gamma', x: C$ is legal (for x fresh).

LEMMA 3.6. *If* Γ_1 *and* Γ_2 *are legal contexts and* $\mathrm{FV}(\Gamma_1) \cap \mathrm{FV}(\Gamma_2) = \emptyset$ *then* Γ_1, Γ_2 *is a legal context.*

PROOF. By induction on the length of Γ_2. If $\Gamma_2 \equiv \langle\,\rangle$ the result is obvious.

If $\Gamma_2 \equiv \Gamma_3, x: A$, then Γ_1, Γ_3 is legal by the induction hypothesis and, as by Lemma 3.4 $\Gamma_3 \vdash A: s$ for some $s \in \mathcal{S}$, we have by Thinning $\Gamma_1, \Gamma_3 \vdash A: s$ and by the Start Rule $\Gamma_1, \Gamma_2 \vdash x: A$. Thus Γ_1, Γ_2 is legal. ⊣

The Generation Lemma of Barendregt [2] is sharpened in the following way.

LEMMA 3.7 (The Sharpened Generation Lemma). *If* $\Gamma \vdash P: B$ *then*

(i) $P \equiv c \in \mathscr{C} \Longrightarrow (c: B) \in \mathscr{A} \vee \exists B'\,[B =_\beta B' \ \&\ (c: B') \in \mathscr{A} \ \&\ \exists s\,[\Gamma \vdash B: s]]$.

(ii) $P \equiv x \in \mathrm{V} \Longrightarrow (x: B) \in \Gamma \vee \exists B'\,[B =_\beta B' \ \&\ (x: B') \in \Gamma \ \&\ \exists s\,[\Gamma \vdash B: s]]$.

(iii) $P \equiv (\Pi x: A.C) \Longrightarrow \exists s_1, s_2, s_3\,[\Gamma \vdash A: s_1 \ \&\ \Gamma, x: A \vdash C: s_2 \ \&\ (s_1, s_2, s_3) \in \mathscr{R} \ \&\ [B \equiv s_3 \vee [B =_\beta s_3 \ \&\ \exists s\,[\Gamma \vdash B: s]]]]$.

(iv) $P \equiv (\lambda x: A.M) \Longrightarrow \exists C, s_3\,[\Gamma \vdash (\Pi x: A.C): s_3 \ \&\ \Gamma, x: A \vdash M: C \ \&\ [B \equiv \Pi x: A.C \vee [B =_\beta \Pi x: A.C \ \&\ \exists s\,[\Gamma \vdash B: s]]]]$.

(v) $P \equiv MN \Longrightarrow \exists A, C\,[\Gamma \vdash M: (\Pi x: A.C) \ \&\ \Gamma \vdash N: A \ \&\ [B \equiv C[x := N] \vee [B =_\beta C[x := N] \ \&\ \exists s\,[\Gamma \vdash B: s]]]]$.

In each case the derivations of the judgements of the form $\Gamma \vdash Q : D$ *have shorter length than that of* $\Gamma \vdash P : B$, *where the length of a derivation is the total number of steps in the derivation.* (*The two judgements* $\Gamma, x : A \vdash C : s_2$ *in* (iii) *and* $\Gamma, x : A \vdash M : C$ *in* (iv) *may not have shorter derivations.*)

PROOF. We only consider case (iii). The other cases can be treated similarly. We distinguish three cases corresponding to the last step in the derivation of $\Gamma \vdash \Pi x : A.C : B$.

Case product rule. This is clearly OK.

Case weakening rule. $\Gamma = \Gamma^-, y : D$ and $\Gamma^-, y : D \vdash \Pi x : A.C : B$ is a direct consequence of $\Gamma^- \vdash \Pi x : A.C : B$ & $\Gamma^- \vdash D : s$.

By the induction hypothesis applied to $\Gamma^- \vdash \Pi x : A.C : B$ we get $\exists s_1, s_2, s_3\, [\Gamma^- \vdash A : s_1$ & $\Gamma^-, x : A \vdash C : s_2$ & $(s_1, s_2, s_3) \in \mathscr{R}$ & $[B \equiv s_3 \vee [B =_\beta s_3$ & $\exists s'\, [\Gamma^- \vdash B : s']]]]$. Moreover $\Gamma^- \vdash A : s_1$ by a shorter derivation than that of $\Gamma^- \vdash \Pi x : A.C : B$. Hence by weakening $\Gamma^-, y : D \vdash A : s_1$ by a shorter derivation than that of $\Gamma^-, y : D \vdash \Pi x : A.C : B$ and similarly for $\Gamma^-, y : D \vdash B : s'$. From $\Gamma^-, x : A \vdash C : s_2$ & $\Gamma^- \vdash D : s$ we get by weakening $\Gamma^-, x : A, y : D \vdash C : s_2$. Finally $\Gamma^-, y : D, x : A \vdash C : s_2$ follows by Lemma 3.5 from the legality of $\Gamma^-, y : D, x : A$.

Case conversion rule. $\Gamma \vdash \Pi x : A.C : B$ is a direct consequence of $\Gamma \vdash \Pi x : A.C : D$ & $\Gamma \vdash B : s$ & $D =_\beta B$. Now the result follows easily from the induction hypothesis applied to $\Gamma \vdash \Pi x : A.C : D$. ⊣

REMARKS.

(i) The sharpening of the Sharpened Generation Lemma with respect to the Generation Lemma of Barendregt [2] consists on the one hand in the addition of $\exists s\, [\Gamma \vdash B : s]$ in (i)–(v). This is not new, it occurs for example in Kamareddine and Nederpelt [18]. The assertions on the lengths of derivations are new and are useful in proofs by induction on derivations. In fact we used it in the proofs of Theorem 8.1 and Theorem 9.6.

(ii) The judgement $\Gamma, x : A \vdash C : s_2$ in (iii) may not have a shorter derivation than that of $\Gamma \vdash \Pi x : A.C : B$ (and similarly for $\Gamma, x : A \vdash M : C$ in (iv)). This is caused by the fact that possibly the last step in the derivation of $\Gamma \vdash \Pi x : A.C : B$ was not by the product rule but by the weakening rule. The following is an example for this. In the PTS with

$$\mathscr{S} = \{s_1, s_2, s_3\}, \quad \mathscr{A} = \{s_1 : s_2, s_2 : s_3\} \quad \text{and} \quad \mathscr{R} = \{(s_2, s_3, s_3)\}$$

the derivation

$$\dfrac{\dfrac{\dfrac{\vdash s_1 : s_2 \quad \vdash s_2 : s_3}{\vdash s_1 : s_2 \quad x : s_1 \vdash s_2 : s_3}}{\vdash \Pi x : s_1.s_2 : s_3} \qquad \vdash s_1 : s_2}{y : s_1 \vdash \Pi x : s_1.s_2 : s_3}$$

has length 3. Now the last step in a shortest derivation of $y : s_1, x : s_1 \vdash s_2 : s_3$ is necessarily a weakening step, hence a shortest derivation is

$$\dfrac{\dfrac{\vdash s_2 : s_3 \quad \vdash s_1 : s_2}{y : s_1 \vdash s_2 : s_3} \qquad \dfrac{\vdash s_1 : s_2 \quad \vdash s_1 : s_2}{y : s_1 \vdash s_1 : s_2}}{y : s_1, x : s_1 \vdash s_2 : s_3}$$

which has also length 3. (If one defines the length of a derivation as the number of steps in a longest branch, then the above is not a counter-example anymore, but then one easily finds another counter-example.)

By the (Sharpened) Generation Lemma one easily proves:

LEMMA 3.8 (Correctness of Types Lemma).

$$\Gamma \vdash M : A \Longrightarrow \exists s\, [A \equiv s \vee \Gamma \vdash A : s].$$

The following lemma can be useful sometimes:

LEMMA 3.9.

$$\Gamma =_\beta \Gamma' \;\&\; \Gamma' \textit{ is legal } \;\&\; \Gamma \vdash M : A \Longrightarrow \Gamma' \vdash M : A.$$

PROOF. By induction on the derivation of $\Gamma \vdash M : A$. ⊣

DEFINITION 3.10. A PTS $\lambda S = \lambda(\mathcal{S}, \mathcal{A}, \mathcal{R})$ is *singly sorted* if

(i) $(c : s_1), (c : s_2) \in \mathcal{A} \Longrightarrow s_1 = s_2$.
(ii) $(s_1, s_2, s_3), (s_1, s_2, s_3') \in \mathcal{R} \Longrightarrow s_3 = s_3'$.

All well-known PTSs are singly sorted. Terms in singly sorted PTSs have unique types:

LEMMA 3.11 (Uniqueness of Types Lemma). *In a singly sorted PTS*:

$$\Gamma \vdash A : B_1 \;\&\; \Gamma \vdash A : B_2 \Longrightarrow B_1 =_\beta B_2.$$

The following definition, taken from Pollack [22], will also be needed below.

DEFINITION 3.12. $s \in \mathcal{S}$ is a *typed sort* if there is an $s' \in \mathcal{S}$ such that $(s : s') \in \mathcal{A}$.

It follows by (i) of the Generation Lemma that s is a typed sort if and only if there are a pseudoterm A and a context Γ such that $\Gamma \vdash s : A$.

The next definition and three lemmas are needed in Section 8.

DEFINITION 3.13.

(i) $\mathcal{S}_I = \{ s \mid \exists \Gamma, A\, [\Gamma \vdash A : s] \}$ (This is the set of inhabited sorts).
(ii) $\mathcal{S}_1 = \{ s_1 \in \mathcal{S}_I \mid \exists s_2, s_3\, [(s_1, s_2, s_3) \in \mathcal{R}] \}$.
(iii) $\mathcal{S}_3 = \{ s_3 \mid \exists s_1, s_2 \in \mathcal{S}_I\, [(s_1, s_2, s_3) \in \mathcal{R}] \}$.

NOTE. $\mathcal{S}_I = \mathcal{S}$ in all well known PTSs. In fact, a sort in $\mathcal{S}$ but not in $\mathcal{S}_I$ would never occur in a derivable judgement and this would be quite strange.

LEMMA 3.14. *If* $\Gamma \vdash M : s$ *then at least one of*

(i) $M : s \in \mathcal{A}$;
(ii) s *is typed sort*;
(iii) $s \in \mathcal{S}_3$.

PROOF. By induction on the derivation of $\Gamma \vdash M : s$. The only non trivial case is where $\Gamma \vdash M : s$ is obtained by the application rule from

$$\Gamma \vdash P : \Pi x : A.B \;\&\; \Gamma \vdash N : A$$

with $M \equiv PN$ and $s \equiv B[x := N]$. Then $B \equiv s$ or $N \equiv s$ (and $B \equiv x$). In the former case we have by the Generation Lemma $\Gamma, x : A \vdash s : s'$, in the latter case $\Gamma \vdash s : A$, so in either case (ii) holds. ⊣

LEMMA 3.15. *$\mathcal{S}_I$ is the set generated recursively by*

(i) $(c : s) \in \mathscr{A} \vee (s : s') \in \mathscr{A} \Longrightarrow s \in \mathcal{S}_I$;
(ii) $s_1, s_2 \in \mathcal{S}_I \;\&\; (s_1, s_2, s) \in \mathscr{R} \Longrightarrow s \in \mathcal{S}_I$.

PROOF. Let $\mathcal{S}_I'$ be the set generated by (i) and (ii) with $\mathcal{S}_I'$ for $\mathcal{S}_I$.
If $s \in \mathcal{S}_I$ then $\Gamma \vdash A : s$ for some Γ and A and $s \in \mathcal{S}_I'$ follows by Lemma 3.14.
If $s \in \mathcal{S}_I'$ there are 3 cases:
If $(c : s) \in \mathscr{A}$ for some $c \in \mathscr{C}$ then $\vdash c : s$ so $s \in \mathcal{S}_I$.
If $(s : s') \in \mathscr{A}$ for some s' then $\vdash s : s'$ and by a start rule $x : s \vdash x : s$, so $s \in \mathcal{S}_I$.
If $(s_1, s_2, s) \in \mathscr{R}$ where $s_1, s_2 \in \mathcal{S}_I$ we have for some Γ_1, Γ_2, B and C $\Gamma_1 \vdash B : s_1$ & $\Gamma_2 \vdash C : s_2$, where we can assume $\mathrm{FV}(\Gamma_1) \cap \mathrm{FV}(\Gamma_2) = \emptyset$. We then have, using Lemma 3.6 and the Thinning Lemma, $\Gamma_1, \Gamma_2 \vdash B : s_1$ & $\Gamma_1, \Gamma_2 \vdash C : s_2$. By a weakening rule we get $\Gamma_1, \Gamma_2, x : B \vdash C : s_2$ and so $\Gamma_1, \Gamma_2 \vdash (\Pi x : B.C) : s$, hence $s \in \mathcal{S}_I$. ⊣

Combining Lemma 3.14 and the Correctness of Types Lemma 3.8 we get:

LEMMA 3.16. *If $\Gamma \vdash M : A$ then at least one of*

(i) $A \in \mathscr{C}$
(ii) $\exists s\, [\Gamma \vdash A : s \;\&\; (s \text{ is a typed sort or } s \in \mathcal{S}_3)]$.

§4. Set-modified PTSs. It is well known that in a PTS-judgement $\vdash M : A$, the type A can be interpreted as a formula of the $(\forall \rightarrow)$-fragment of (a possibly higher order) intuitionistic predicate logic with $\Pi x : A.B$ representing $A \rightarrow B$ if $x \notin \mathrm{FV}(B)$ and $(\forall x : A)\, B$ otherwise. M represents a natural deduction style proof of A. Similarly a PTS-judgement $x_1 : A_1, \ldots, x_n : A_n \vdash M : A$ represents the deduction $A_1, \ldots, A_n \vdash A$, where again $A_1, \ldots, A_n, A$ are formulas as above and M is a proof of A subject to the hypotheses $A_1, \ldots, A_n$ which are coded by $x_1, \ldots, x_n$. This isomorphism is however, in one sense, not fully natural. In intuitionistic logic $A_1, \ldots, A_n$ represents a set while in a PTS $x_1 : A_1, \ldots, x_n : A_n$ forms a sequence.

In this section we introduce Set-modified PTSs in which contexts are sets. This is done by allowing more liberal axioms and a more liberal start rule. Also the conversion rule is slightly generalised. This change also represents one step in the linking of PTSs and ICLs.

Similar changes, leading to a system TOC2 of the Calculus of Constructions, were considered in Seldin [23].

DEFINITION 4.1.

(i) If Δ and Δ' are sets of statements then $\Delta \twoheadrightarrow_\beta \Delta'$ if for all $M : A \in \Delta$ there is an $N : B \in \Delta'$ such that $M \twoheadrightarrow_\beta N$ and $A \twoheadrightarrow_\beta B$ and for each $N : B \in \Delta'$ there is an $M : A \in \Delta$ such that $M \twoheadrightarrow_\beta N$ and $A \twoheadrightarrow_\beta B$.
(ii) $=_\beta$ over sets of statements is the equivalence relation generated by $\twoheadrightarrow_\beta$.

DEFINITION 4.2. A *Set-modified Pure Type System (SPTS)* $\lambda^S S = \lambda^S(\mathcal{S}, \mathscr{A}, \mathscr{R})$ is determined by a triple $(\mathcal{S}, \mathscr{A}, \mathscr{R})$ as for PTSs. The notion of type derivation written as $\Delta \vdash^{\lambda S}_S M : A$ (or just $\Delta \vdash_S M : A$), where Δ is a set of statements, is defined by the following postulates:

(axioms) $\Delta \vdash_S c : s$ if $(c : s) \in \mathscr{A}$;

(start) $\Delta \vdash_S M : A$ if $(M : A) \in \Delta$;

(product) $$\frac{\Delta \vdash_S A : s_1 \quad \Delta, x : A \vdash_S B : s_2}{\Delta \vdash_S (\Pi x : A.B) : s_3} \quad \text{if } x \notin \mathrm{FV}(\Delta), (s_1, s_2, s_3) \in \mathscr{R};$$

(abstraction) $$\frac{\Delta, x : A \vdash_S M : B \quad \Delta \vdash_S (\Pi x : A.B) : s}{\Delta \vdash_S (\lambda x : A.M) : (\Pi x : A.B)} \quad \text{if } x \notin \mathrm{FV}(\Delta, A);$$

(application) $$\frac{\Delta \vdash_S M : (\Pi x : A.B) \quad \Delta \vdash_S N : A}{\Delta \vdash_S (MN) : B[x := N]};$$

(gen conv) $$\frac{\Delta \vdash_S M : A \quad \Delta =_\beta \Delta', M =_\beta N, A =_\beta B}{\Delta' \vdash_S N : B}.$$

NOTES.

(i) The product and abstraction rules above have restrictions $x \notin \mathrm{FV}(\Delta)$ and $x \notin \mathrm{FV}(\Delta, A)$. The corresponding $x \notin \mathrm{FV}(\Gamma)$ and $x \notin \mathrm{FV}(\Gamma, A)$ are derivable in PTSs.

(ii) The statements that are the elements of Δ are not restricted in any way, for example the start rule can derive $M : A$ where M need not even be β-equal to a variable.

(iii) The generalised conversion rule is not admissible in PTSs. For example for $(s : s') \in \mathscr{A}$ we have $x : s \vdash x : s$ but not $(\lambda y : A.x)y : s \vdash x : s$ (which is not even a PTS-judgement), nor $x : s \vdash x : (\lambda y : x.y)s$. Both judgements are valid for all SPTSs. The latter example shows that (gen conv) even with only $A =_\beta B$, but without $\Gamma \vdash B : s$, is not admissible for PTSs. We note that (gen conv) gives subject reduction (which is valid for PTSs, cf. Lemma 3.3) and subject expansion (which, in general, is not valid for PTSs).

(iv) The Church Rosser Theorem for contexts 2.7 (ii) can easily be extended to sets of statements.

As we have $\Delta \vdash_S M : A$ for any $M : A \in \Delta$ we require a different definition of legal context in an SPTS than the one for a PTS. It is for these legal contexts that we can prove PTSs and SPTSs, with the same specification, equivalent.

DEFINITION 4.3. A set Δ is said to be *S-legal* in an SPTS λS if $\Delta =_\beta \{x_1 : A_1, \ldots, x_n : A_n\}$ and

(i) $(\forall i, j)\,[1 \leq i < j \leq n \Longrightarrow x_i \neq x_j]$
(ii) $(\forall i)\,[1 \leq i \leq n \Longrightarrow (\exists s_i \in \mathscr{S})\,[x_1 : A_1, \ldots, x_{i-1} : A_{i-1} \vdash_S A_i : s_i]]$
(iii) $(\forall i)\,[1 \leq i \leq n \Longrightarrow x_i, \ldots, x_n \notin \mathrm{FV}(A_i)]$.

Note that (i)–(iii), with $\vdash_S$ replaced by $\vdash$ hold for a legal PTS context $x_1 : A_1, \ldots, x_n : A_n$.

Several of the usual PTS properties can be derived for SPTSs using the equivalence property proved in the next section, but only where the Δ in any judgement is legal.

Most of these properties can be proved directly for a general set Δ. The one property that is needed in the proof of equivalence is the following.

LEMMA 4.4 (The Thinning Lemma for SPTSs). *If* $\Delta \vdash_S M : A$ *and* $\Delta \subseteq \Delta'$ *then* $\Delta' \vdash_S M : A$.

PROOF. By an easy induction on the derivation of $\Delta \vdash_S M : A$. ⊣

§5. The equivalence between PTSs and SPTSs.

DEFINITION 5.1. For Γ a PTS-context $S(\Gamma)$ denotes the set consisting of the statements of Γ.

Two further lemmas are required to prove the equivalence theorem.

LEMMA 5.2. *If*

$$\Delta \vdash_S M : A \tag{1}$$

$\Delta =_\beta S(\Gamma)$, Γ *a legal context for the PTS with the same specification, then there exist pseudoterms* M' *and* A' *such that* $M =_\beta M'$, $A =_\beta A'$ *and*

$$\Gamma \vdash M' : A'. \tag{2}$$

PROOF. By induction on the derivation of (1).

(*Case axiom*) Now $M : A \in \mathscr{A}$. In this case $M \equiv M'$, $A \equiv A'$ and (2) follows by the Start Lemma for PTSs.

(*Case start*) Now $M : A \in \Delta$. As $\Delta =_\beta S(\Gamma)$ there is an $M'' : A'' \in \Gamma$ such that $M =_\beta M''$ and $A =_\beta A''$. By the Start Lemma for PTSs we get $\Gamma \vdash M'' : A''$.

(*Case product*) $M \equiv \Pi x : B.C$, $A \equiv s_3$ and $(s_1, s_2, s_3) \in \mathscr{R}$ and (1) is obtained from

$$\Delta \vdash_S B : s_1 \tag{3}$$

$$\Delta, x : B \vdash_S C : s_2. \tag{4}$$

By the induction hypothesis and Lemma 3.3

$$\Gamma \vdash B' : s_1 \tag{5}$$

where $B =_\beta B'$. So $\Gamma, x : B'$ is legal. As $\Delta, x : B =_\beta S(\Gamma, x : B')$ we have by (4), the induction hypothesis and Lemma 3.3

$$\Gamma, x : B' \vdash C' : s_2 \tag{6}$$

where $C =_\beta C'$. By the product rule we get from (5) and (6)

$$\Gamma \vdash (\Pi x : B'.C') : s_3.$$

This is (2) with $M' \equiv \Pi x : B'.C'$ and $A \equiv A'$.

The *Cases application, abstraction and generalised conversion* can be treated in a similar way. Now also Church Rosser is needed. In the *Case abstraction* moreover the Generation Lemma 3.7 for PTSs is needed in order to derive from $\Gamma \vdash \Pi x : B'.C' : s$ that $\Gamma, x : B' \vdash C' : s'$ for some s' and hence $\Gamma, x : B'$ is legal. ⊣

LEMMA 5.3. *If* Δ *is S-legal for a given SPTS, there is a context* Γ, *legal for the PTS with the same specification, such that* $\Delta =_\beta S(\Gamma)$.

PROOF. By induction on the number n in Definition 4.3.

If $n = 1$ then $\Delta =_\beta \{x: A\}$, where $\vdash_S A: s$ and $x \notin \mathrm{FV}(A)$. By Lemma 5.2 and Subject Reduction there is an A' such that $A =_\beta A'$ and $\vdash A': s$. Thus $\Delta =_\beta \{x: A'\}$ and, as by a start rule $x: A' \vdash x: A'$, we have that $x: A'$ is legal.

If $n > 1$ we have $x_1, \ldots, x_n$ and $A_1, \ldots, A_n$ such that $\Delta =_\beta \{x_1: A_1, \ldots, x_n: A_n\}$, where 4.3 (i), (ii) and (iii) hold. It follows that $\{x_1: A_1, \ldots, x_{n-1}: A_{n-1}\}$ is also S-legal and, by the induction hypothesis, that there is a legal context Γ^-, such that $\{x_1: A_1, \ldots, x_{n-1}: A_{n-1}\} =_\beta S(\Gamma^-)$. Now by Lemma 5.2, Definition 4.3 (ii) with $i = n$ and Subject Reduction we have $\Gamma^- \vdash A'_n: s_n$ where $A_n =_\beta A'_n$. So $\Gamma^-, x_n: A'_n$ is legal. As $\Delta =_\beta S(\Gamma^-, x_n: A'_n)$, we have the required result. ⊣

THEOREM 5.4 (The Equivalence Theorem for PTSs and SPTSs). *For any PTS and SPTS with the same specification*

(i) $\Gamma \vdash M: A \Longrightarrow S(\Gamma)$ *is S-legal* & $S(\Gamma) \vdash_S M: A$.

(ii) Δ *is S-legal* & $\Delta \vdash_S M: A \Longrightarrow (\exists \Gamma, M', A')[\Delta =_\beta S(\Gamma)$ & $M =_\beta M'$ & $A =_\beta A'$ & $\Gamma \vdash M': A']$.

PROOF.

(i) We let $\Gamma \equiv \langle x_1: A_1, \ldots, x_n: A_n\rangle$ and proceed by induction on the derivation of

$$\Gamma \vdash M: A. \tag{1}$$

(*Case axiom*) Now $\Gamma \equiv \langle\,\rangle$, $S(\Gamma) = \emptyset$ and is S-legal and $S(\Gamma) \vdash_S M: A$.

(*Case start*) $\Gamma \equiv \Gamma^-, x_n: A_n$, $M \equiv x_n$, $A \equiv A_n$, and (1) is obtained from $\Gamma^- \vdash A_n: s$. By the induction hypothesis we have: $S(\Gamma^-)$ is S-legal and $S(\Gamma^-) \vdash_S A_n: s$. Also $x_n \neq x_i$ and $x_n \notin \mathrm{FV}(A_i)$ for $1 \leq i < n$, so $S(\Gamma)$ is S-legal. $M: A \in S(\Gamma)$, hence $S(\Gamma) \vdash_S M: A$.

(*Case weakening*) $\Gamma \equiv \Gamma^-, x_n: A_n$ and (1) is obtained from $\Gamma^- \vdash M: A$, $\Gamma^- \vdash A_n: s$. We have as above that $S(\Gamma)$ is legal. By the induction hypothesis we have $S(\Gamma^-) \vdash_S M: A$ from which $S(\Gamma) \vdash_S M: A$ follows by the Thinning Lemma for SPTSs.

(*Other cases*) If (1) is obtained by one of the other rules, we find by the induction hypothesis applied to one of the premises from which (1) is obtained that $S(\Gamma)$ is S-legal. In each case $S(\Gamma) \vdash_S M: A$ follows when the induction hypothesis is applied to the premise.

(ii) This follows directly from Lemmas 5.2 and 5.3. ⊣

REMARKS.

(i) From the Equivalence Theorem follows directly: For any PTS and SPTS with the same specification

$$\Delta \text{ is S-legal} \iff \Delta =_\beta S(\Gamma) \text{ for some legal } \Gamma.$$

(ii) By Church Rosser for PTSs and SPTSs, Subject Reduction for PTSs and General Conversion for SPTSs we may replace $=_\beta$ by $\twoheadrightarrow_\beta$ in 4.3, 5.2 and 5.3.

LEMMA 5.5 (The Generation Lemma for SPTSs). *If* Δ *is S-legal and* $\Delta \vdash_S P: B$ *then*

(i) $P =_\beta c \in \mathscr{C} \Longrightarrow \exists B'\,[B =_\beta B'$ & $(c: B') \in \mathscr{A}]$.

(ii) $P =_\beta x \in \mathrm{V} \Longrightarrow \exists B', M\,[x =_\beta M$ & $B =_\beta B'$ *where* $M: B' \in \Delta]$.

(iii) $P =_\beta (\Pi x\colon A.C) \Longrightarrow \exists s_1, s_2, s_3\ [\Delta \vdash_S A\colon s_1\ \&\ \Delta, x\colon A \vdash_S C\colon s_2\ \&\ (s_1, s_2, s_3) \in \mathscr{R}\ \&\ B =_\beta s_3]$.

(iv) $P =_\beta (\lambda x\colon A.M) \Longrightarrow \exists C, s\ [\Delta \vdash_S (\Pi x\colon A.C)\colon s\ \&\ \Delta, x\colon A \vdash_S M\colon C\ \&\ B =_\beta \Pi x\colon A.C]$.

(v) $P =_\beta MN\ \&\ M \neq_\beta \lambda x\colon A'.M'$ *for any* A' *and* $M' \Longrightarrow \exists A, C\ [\Delta \vdash_S M\colon (\Pi x\colon A.C)\ \&\ \Delta \vdash_S N\colon A\ \&\ B =_\beta C[x := N]]$.

PROOF. This follows immediately from the (Sharpened) Generation Lemma for PTSs by the Equivalence Theorem 5.4. Note that the sharpening is lost. We were not able to find a direct proof of this Generation Lemma, without using the Equivalence Theorem. Moreover note that the lemma does not cover the case: $P =_\beta MN\ \&\ M =_\beta \lambda x\colon A'.M'$ ⊣

§6. Abstraction-modified PTSs. In PTSs the start and weakening rules which introduce only particular hypotheses to the left of the $\vdash$ partly fulfill the function of the formation rules of predicate logic which ensure that only well formed formulas are used. In logic, given that the hypotheses are well formed, the $\rightarrow$ introduction rule can be freely applied. For PTSs however, the corresponding (abstraction) rule requires an extra condition on the new formula being constructed. As a result, again, the formulas as types isomorphism is not fully natural for PTSs. A fully natural PTS-counterpart to the $\rightarrow$ introduction rule would be

$$\frac{\Gamma, x\colon A \vdash M\colon B}{\Gamma \vdash (\lambda x\colon A.M)\colon (\Pi x\colon A.B)}\ .$$

The system that results is more or less the system with correctness relation $\vdash_0$ of van Benthem Jutting, Mc Kinna and Pollack [4] and the system $\lambda^\omega(S)$ of Severi [24]. However as we shall see in Theorem 8.5 it is only for the PTS $\lambda*$ that this rule is equivalent to the standard PTS abstraction rule. For other PTSs the new rule is not valid. To have the new rule valid for PTSs we need, for some $s_3 \in \mathscr{S}$: $\Gamma \vdash (\Pi x\colon A.B)\colon s_3$ which requires

(a) $$\Gamma, x\colon A \vdash B\colon s_2$$

(b) $$\Gamma \vdash A\colon s_1,$$

where $(s_1, s_2, s_3) \in \mathscr{R}$. By the Correctness of Types Lemma (3.8) we have (a) for some s_2, provided B is not a sort or is a typed sort. We also have (b) for some s_1, by Lemmas 3.4 and 3.5, but we are not guaranteed that for the particular s_1 and s_2 we have $(s_1, s_2, s_3) \in \mathscr{R}$. We modify the abstraction rule as follows:

DEFINITION 6.1 (The Modified Abstraction Rule).

$$\frac{\Gamma, x\colon A \vdash M\colon B \quad \Gamma \vdash A\colon s \quad (+) \text{ holds for } s \text{ and } B}{\Gamma \vdash (\lambda x\colon A.M)\colon (\Pi x\colon A.B)}$$

where '$(+)$ holds for s and B' denotes

$$(\exists s_2, s_3)\ [(s, s_2, s_3) \in \mathscr{R}\ \&\ (B \in \mathscr{C} \Longrightarrow (B\colon s_2) \in \mathscr{A})].$$

The restriction $\Gamma \vdash A\colon s$ is similar to that used for ICLs (where also Γ is an arbitrary set). For ICLs and PTSs that are equivalent $(+)$ should be derivable.

Of the nine standard PTSs considered in Theorem 8.5 below, one satisfies (+), for three (+) translates to "B is not a sort" and for the other five to "B is a typed sort or not a sort."

Pollack [22] and van Benthem Jutting, Mc Kinna and Pollack [4] have a similar condition:

$$(\exists s_2, s_3)\,[(s, s_2, s_3) \in \mathscr{R} \;\&\; (B \in \mathscr{C} \Longrightarrow (\exists s_4)[(B : s_4) \in \mathscr{A}]].$$

Let's denote this by (+′). Our condition (+) is a little bit stronger because in (+) s_2 and s_4 are identified. We will show later that our system, based on (+), which we define below, allows more PTSs and corresponding modified PTSs to be proved equivalent.

The version TOC0 of the Calculus of Constructions of Seldin [23] has this abstraction rule but without condition (+).

DEFINITION 6.2. An *Abstraction modified PTS* (APTS) $\lambda^A S = \lambda^A(\mathscr{S}, \mathscr{A}, \mathscr{R})$ is a PTS with the given specification and the PTS postulates except the abstraction rule which is replaced by the Modified Abstraction Rule 6.1. Type derivations will be written as $\Gamma \vdash^{\lambda S}_{\mathrm{A}} M : A$ (or just $\Gamma \vdash_{\mathrm{A}} M : A$).

We will call a PTS with a modified abstraction rule with (+′) instead of (+) a BPTS.

DEFINITION 6.3. A context Γ is said to be A-legal if, for some M and B, $\Gamma \vdash_{\mathrm{A}} M : B$.

§7. Some lemmas for APTSs. The following lemmas 7.1-7.4 are proved in the same way as the corresponding lemmas for PTSs.

LEMMA 7.1 (The Start Lemma for APTSs). *If Γ is an A-legal context then*

(i) $(c : s) \in \mathscr{A} \Longrightarrow \Gamma \vdash_{\mathrm{A}} c : s$;

(ii) $(x : A) \in \Gamma \Longrightarrow \Gamma \vdash_{\mathrm{A}} x : A$.

LEMMA 7.2. *If $x_1 : A_1, \ldots, x_n : A_n \vdash_{\mathrm{A}} M : A$ then for each i, $1 \leq i \leq n$, there is an $s_i \in \mathscr{S}$ such that the derivation of $x_1 : A_1, \ldots, x_n : A_n \vdash_{\mathrm{A}} M : A$ contains a derivation of $x_1 : A_1, \ldots, x_{i-1} : A_{i-1} \vdash_{\mathrm{A}} A_i : s_i$.*

LEMMA 7.3 (The Thinning Lemma for APTSs). *If Γ and Γ' are A-legal contexts and $\Gamma \subseteq \Gamma'$ then*

$$\Gamma \vdash_{\mathrm{A}} M : A \Longrightarrow \Gamma' \vdash_{\mathrm{A}} M : A.$$

LEMMA 7.4 (The Sharpened Generation Lemma for APTSs). *This is the same as the Sharpened Generation Lemma* 3.7 *for PTSs, with $\vdash$ replaced by $\vdash_{\mathrm{A}}$, except the abstraction case* (iv)

$$\begin{aligned} P \equiv (\lambda x : A.M) \Longrightarrow \exists C, s_1, s_2, s_3\, [\Gamma, x : A \vdash_{\mathrm{A}} M : C \;\&\; \Gamma \vdash_{\mathrm{A}} A : s_1 \\ \&\; (s_1, s_2, s_3) \in \mathscr{R} \;\&\; [C \in \mathscr{C} \Longrightarrow (C : s_2) \in \mathscr{A})] \\ \&\; [B \equiv \Pi x : A.C \vee [B =_\beta \Pi x : A.C \;\&\; \exists s\, [\Gamma \vdash_{\mathrm{A}} B : s]]]]. \end{aligned}$$

Other lemmas, which hold for PTSs and SPTSs, however, do not hold as is shown below.

LEMMA 7.5. *The Subject Reduction and Substitution Theorems do not hold for APTSs.*

PROOF. Consider an APTS with $\mathcal{S} = \{s_1, s_2, s_3\}$, $\mathcal{A} = \{s_1 : s_2, s_2 : s_3\}$ and $\mathcal{R} = \{(s_2, s_3, s_3), (s_3, s_3, s_3)\}$. We have

$$y : s_2, x : y \vdash_A x : y, \quad y : s_2 \vdash_A y : s_2$$

and as $(s_2, s_3, s_3) \in \mathcal{R}$, $(+)$ holds for s_2 and y, so

$$y : s_2 \vdash_A \lambda x : y.x : \Pi x : y.y. \tag{1}$$

Also $\vdash_A s_2 : s_3$ and as $(s_3, s_3, s_3) \in \mathcal{R}$, $(+)$ holds for s_3 and $\Pi x : y.y$, so $\vdash_A (\lambda y : s_2.\lambda x : y.x) : (\Pi y : s_2.\Pi x : y.y)$. Now

$$\vdash_A s_1 : s_2 \tag{2}$$

so by application $\vdash_A (\lambda y : s_2.\lambda x : y.x)s_1 : (\Pi x : s_1.s_1)$. If Subject Reduction holds or, by (1) and (2), if the Substitution Theorem holds, we have

$$\vdash_A (\lambda x : s_1.x) : (\Pi x : s_1.s_1).$$

By (iv), (iii) and (i) of the Sharpened Generation Lemma for APTSs this requires

$$\exists s_k, s_l, s_m \, [(s_k, s_l, s_m) \in \mathcal{R}, \quad (s_1 : s_k) \in \mathcal{A}, \quad (s_1 : s_l) \in \mathcal{A}]$$

which does not hold. ⊣

Note that in the same APTS we can prove:

$$\vdash_A (\lambda x : s_1.x) : (\Pi x : s_1.(\lambda z : s_2.s_1)s_1).$$

This also shows that even with the restriction $(+)$ in the APTS Abstraction Rule, not all PTSs and APTSs are equivalent. We will show below that those whose specification satisfies a condition $(*)$ are equivalent. Moreover we show that for a singly sorted PTS and an APTS with the same specification $(*)$ is not only a sufficient condition but also a necessary condition for the equivalence.

§8. Equivalences between PTSs and APTSs. We will now derive results connecting $\vdash_A$ with $\vdash$.

THEOREM 8.1. *For any PTS and an APTS with the same specification*

$$\Gamma \vdash P : C \Longrightarrow \Gamma \vdash_A P : C.$$

PROOF. By induction on the derivation of $\Gamma \vdash P : C$. All cases are obvious except where $\Gamma \vdash P : C$ comes by the abstraction rule from

$$\Gamma, x : A \vdash M : B \tag{1}$$

and

$$\Gamma \vdash (\Pi x : A.B) : s_3 \tag{2}$$

where $P \equiv \lambda x : A.M$ and $C \equiv \Pi x : A.B$. By (2) and the Sharpened Generation Lemma there is a rule (s_1, s_2, s_3) such that

$$\Gamma \vdash A : s_1 \tag{3}$$

and $\Gamma, x : A \vdash B : s_2$, where the derivation of (3) is shorter than that of (2). If $B \in \mathcal{C}$ then the Generation Lemma gives $B : s_2 \in \mathcal{A}$. Hence $(+)$ holds for B and s_1. By the induction hypothesis applied to (1) and (3) we have

$$\Gamma, x : A \vdash_A M : B, \ \& \ \Gamma \vdash_A A : s_1$$

which, given $(+)$, gives $\Gamma \vdash_A P : C$. ⊣

REMARK. The reversed arrow $\Leftarrow$ in the theorem above does not hold in general. And worse, the Substitution Lemma and Subject Reduction do not hold in general for APTS's. However we will show that the reversed arrow $\Leftarrow$ (and hence the Substitution Lemma and Subject Reduction) do hold for a certain subclass of specifications.

The abstraction rule mentioned at the start of section 6 (the one without $(+)$) does make for 'nice' systems in this sense, but they are not equivalent to PTSs.

The failure of the Subject Reduction Theorem for APTSs shows that $\Gamma \vdash_A P: C$ can only imply $\Gamma \vdash P: C$ under certain conditions. Inspired by Lemma 3.16 we define:

DEFINITION 8.2. The condition $(*)$ is defined as

$$\forall s_1 \in \mathcal{S}_1\ \forall s_2 \in \mathcal{S}\ [(s_2 \text{ is a typed sort or } s_2 \in \mathcal{S}_3) \Longrightarrow \exists s_3\ [(s_1, s_2, s_3) \in \mathcal{R}]].$$

NOTE. 's_2 is a typed sort or $s_2 \in \mathcal{S}_3$' in $(*)$ implies that s_2 is inhabited by a $B \notin \mathcal{C}$, because if s_2 is a typed sort then, by the start rule, we can take a variable x as B, and if $s_2 \in \mathcal{S}_3$ then s_2 is inhabited by a product.

THEOREM 8.3. *For a PTS and an APTS with the same specification and such that* $(*)$ *holds*

$$\Gamma \vdash_A P: C \Longrightarrow \Gamma \vdash P: C.$$

PROOF. By induction on the derivation of $\Gamma \vdash_A P: C$. The only non-trivial case is where $\Gamma \vdash_A P: C$ is obtained by the abstraction rule from

$$\Gamma, x: A \vdash_A M: B \ \&\ \Gamma \vdash_A A: s$$

where $(+)$ holds for s and B, $P \equiv \lambda x: A.M$ and $C \equiv \Pi x: A.B$. By the induction hypothesis we have

$$\Gamma, x: A \vdash M: B \ \&\ \Gamma \vdash A: s.$$

We only need to show

$$\exists s_3\ [\Gamma \vdash \Pi x: A.B: s_3].$$

Lemma 3.16 applied to $\Gamma, x: A \vdash M: B$ yields that we have at least one of

(i) $B \in \mathcal{C}$,

(ii) $\Gamma, x: A \vdash B: s_2$ & (s_2 is a typed sort or $s_2 \in \mathcal{S}_3$).

In case (i) we get from $(+)$

$$(\exists s_2, s_3)\ [(s, s_2, s_3) \in \mathcal{R}\ \&\ \Gamma, x: A \vdash B: s_2],$$

so $\Gamma \vdash \Pi x: A.B: s_3$.

In case (ii) we get from $(*)$ $\exists s_3\ [(s, s_2, s_3) \in \mathcal{R}]$ (note that $s \in \mathcal{S}_1$ by $(+)$), so $\Gamma \vdash \Pi x: A.B: s_3$. ⊣

PROPOSITION 8.4. *For a singly sorted PTS and an APTS with the same specification*

$$\Gamma \vdash_A P: C \Longrightarrow \Gamma \vdash P: C \text{ for all } \Gamma, P \text{ and } C$$

if and only if $(*)$ *holds.*

PROOF. The if part we have in the theorem. For the only if part assume that

$$\Gamma \vdash_{\mathrm{A}} P\colon C \Longrightarrow \Gamma \vdash P\colon C \text{ for all } \Gamma, P \text{ and } C$$

and also

(1) $$s_1 \in \mathcal{S}_1,\ s_2 \in \mathcal{S},\ s_2 \text{ a typed sort or } s_2 \in \mathcal{S}_3.$$

We must prove: $\exists s_3\,[(s_1, s_2, s_3) \in \mathscr{R}]$. From (1) we get that s_2 is inhabited by a $B \notin \mathscr{C}$. So there are Γ_1, Γ_2, A and B such that

$$\Gamma_1 \vdash A\colon s_1 \ \&\ \Gamma_2 \vdash B\colon s_2 \ \&\ B \notin \mathscr{C}.$$

We can assume $\mathrm{FV}(\Gamma_1) \cap \mathrm{FV}(\Gamma_2) = \emptyset$. As in the proof of Lemma 3.15 we can prove then $\Gamma_1, \Gamma_2 \vdash A\colon s_1$ and $\Gamma_1, \Gamma_2 \vdash B\colon s_2$. Letting $\Gamma = \Gamma_1, \Gamma_2$ we then have by Theorem 8.1

$$\Gamma \vdash_{\mathrm{A}} A\colon s_1 \ \&\ \Gamma \vdash_{\mathrm{A}} B\colon s_2.$$

From this we get by the weakening rule

(2) $$\Gamma, y\colon B \vdash_{\mathrm{A}} A\colon s_1$$

and by the weakening rule and the start rule

(3) $$\Gamma, y\colon B, x\colon A \vdash_{\mathrm{A}} y\colon B.$$

We have $s_1 \in \mathcal{S}_1$ and $B \notin \mathscr{C}$, hence $(+)$ is fulfilled for s_1 and B, so we get from (2) and (3) $\Gamma, y\colon B \vdash_{\mathrm{A}} \lambda x\colon A.y\colon \Pi x\colon A.B$. By the hypothesis we then also have $\Gamma, y\colon B \vdash \lambda x\colon A.y\colon \Pi x\colon A.B$. By the Correctness of Types Lemma (3.8) this gives for some $s_3 \in \mathcal{S}$ $\Gamma, y\colon B \vdash \Pi x\colon A.B\colon s_3$ and by the Generation lemma

$$\Gamma, y\colon B \vdash A\colon s_1' \ \&\ \Gamma, y\colon B, x\colon A \vdash B\colon s_2'$$

where $(s_1', s_2', s_3) \in \mathscr{R}$. By weakening we get from $\Gamma \vdash A\colon s_1, \quad \Gamma \vdash B\colon s_2$

$$\Gamma, y\colon B \vdash A\colon s_1 \ \&\ \Gamma, y\colon B, x\colon A \vdash B\colon s_2.$$

If the PTS is singly sorted we have by the Uniqueness of Types Lemma $s_1 \equiv s_1'$ and $s_2 \equiv s_2'$ and so $(s_1, s_2, s_3) \in \mathscr{R}$. Hence $(*)$ holds. ⊣

In the theorem below (s_1, s_2) as an element of $\mathscr{R}$ is short for (s_1, s_2, s_2).

THEOREM 8.5. *The PTSs specified in Figure* 1 *satisfy* $(*)$ *and so are equivalent to the corresponding APTSs. In each case we note the restriction imposed by* $(+)$ *(the values allowed for* s *and those not allowed for* B*).*

Of the PTSs in Figure 1, $\lambda\!\!\rightarrow$ is the λ-calculus of Church [10] and λC that of Coquand and Huet [11]. The APTS-abstraction rule for λC (i.e., with restriction $B \neq \Box$) is exactly the original rule given of Coquand and Huet [11]. λ-AUT-68, λ-AUT-QE and λ-PAL are some of the de Bruijn AUTOMATH systems. Note that Barendregt [2] shows that with the replacement of $*^p$ for $*$ and $\Box^p$ for $\Box$, $\lambda\!\!\rightarrow$ becomes the PTS λPROP which in turn is related to propositional logic. After a similar substitution $\lambda 2$ becomes λPROP2, related to second order logic.

For the other PTSs (such as λP2, $\lambda\omega$, $\lambda\underline{\omega}$, $\lambda\mathrm{P}\underline{\omega}$, λU and λHOL) mentioned in Barendregt [2] and Geuvers [15], there is no simple APTS equivalent as $(*)$ fails.

The part of the system ECC of Luo [19] without pairing and Σ-types could have been formulated (without a change of provable theorems) so that condition $(*)$ is satisfied. Hence that part of ECC is also equivalent to its corresponding APTS.

$\lambda\rightarrow$: $\mathcal{S}=\{*,\Box\}$, $\mathcal{A}=\{*:\Box\}$, $\mathcal{R}=\{(*,*)\}$,
$(+)$: $s=*$, $B\notin\{*,\Box\}$.

λ^{τ}: $\mathcal{S}=\{*\}$, $\mathcal{A}=\{0:*\}$, $\mathcal{R}=\{(*,*)\}$,
$(+)$: $s=*$, $B\neq *$.

$\lambda *$: $\mathcal{S}=\{*\}$, $\mathcal{A}=\{*:*\}$, $\mathcal{R}=\{(*,*)\}$,
$(+)$: empty.

$\lambda 2$: $\mathcal{S}=\{*,\Box\}$, $\mathcal{A}=\{*:\Box\}$, $\mathcal{R}=\{(*,*),(\Box,*)\}$,
$(+)$: $B\notin\{*,\Box\}$.

λP: $\mathcal{S}=\{*,\Box\}$, $\mathcal{A}=\{*:\Box\}$, $\mathcal{R}=\{(*,*),(*,\Box)\}$,
$(+)$: $s=*$, $B\neq\Box$.

$\lambda C=\lambda P\omega$: $\mathcal{S}=\{*,\Box\}$, $\mathcal{A}=\{*:\Box\}$,
$\mathcal{R}=\{(*,*),(*,\Box),(\Box,*),(\Box,\Box)\}$,
$(+)$: $B\neq\Box$.

λ-*AUT-68*: $\mathcal{S}=\{*,\Box,\Delta\}$, $\mathcal{A}=\{*:\Box\}$,
$\mathcal{R}=\{(*,*),(*,\Box,\Delta),(\Box,*,\Delta),(\Box,\Box,\Delta),(*,\Delta),(\Box,\Delta)\}$,
$(+)$: $s\in\{*,\Box\}$, $B\notin\{\Box,\Delta\}$.

λ*AUT-QE*: $\mathcal{S}=\{*,\Box,\Delta\}$, $\mathcal{A}=\{*:\Box\}$,
$\mathcal{R}=\{(*,*),(*,\Box),(\Box,*,\Delta),(\Box,\Box,\Delta),(*,\Delta),(\Box,\Delta)\}$,
$(+)$: $s\in\{*,\Box\}$, $B\notin\{\Box,\Delta\}$.

λ-*PAL*: $\mathcal{S}=\{*,\Box,\Delta\}$, $\mathcal{A}=\{*:\Box\}$,
$\mathcal{R}=\{(*,*,\Delta),(*,\Box,\Delta),(\Box,*,\Delta),(\Box,\Box,\Delta),(*,\Delta),(\Box,\Delta)\}$,
$(+)$: $s\in\{*,\Box\}$, $B\notin\{\Box,\Delta\}$.

FIGURE 1

Pollack [22] and van Benthem Jutting, Mc Kinna and Pollack [4] use a condition stronger than $(*)$ called semi-fullness:

DEFINITION 8.6. A PTS is semi-full if

$$\forall s_1\,[\exists s_2,s_3\,[(s_1,s_2,s_3)\in\mathcal{R}]\Longrightarrow\forall s_2\,\exists s_3\,[(s_1,s_2,s_3)\in\mathcal{R}]].$$

They show that semi-full BPTSs are equivalent to PTSs with the same specification.

For semi-full PTSs $(+)$ and $(+')$ are equivalent and hence so are APTSs and BPTSs. Proposition 8.4, however does not hold for BPTSs. In fact $\lambda\rightarrow$ and $\lambda 2$ are examples of PTSs, which are not semi-full, but satisfy $(*)$, that are equivalent to their corresponding APTSs.

§9. Set-abstraction modified PTSs. We now consider PTSs with both the Set- and Abstraction Modifications. They are very similar to ICL-systems.

DEFINITION 9.1. A Set-Abstraction modified PTS (SAPTS) is a PTS with

(i) All contexts Γ replaced by sets Δ,
(ii) The rules (axioms), (start) and (gen conv) as in 4.2 for SPTSs,
(iii) The Abstraction Rule replaced by the Modified Abstraction Rule 6.1 but with Condition (+) strengthened in the following way: $(\exists s_2, s_3)\,[(s, s_2, s_3) \in \mathscr{R}\ \&\ (\forall B')\,(B =_\beta B' \in \mathscr{C} \Longrightarrow (B' : s_2) \in \mathscr{A})]$.

Type derivation will be written as $\Delta \vdash^{\lambda S}_{SA} M : A$ (or just $\Delta \vdash_{SA} M : A$).

DEFINITION 9.2. A set Δ is said to be SA-legal in an SAPTS if $\Delta =_\beta \{x_1 : A_1, \ldots, x_n : A_n\}$ and

(i) $(\forall i, j)\,(1 \le i < j \le n \Longrightarrow x_i \ne x_j)$;
(ii) $(\forall i)\,(1 \le i \le n \Longrightarrow (\exists s_i \in S)\,(x_1 : A_1, \ldots, x_{i-1} : A_{i-1} \vdash_{SA} A_i : s_i)$;
(iii) $(\forall i)\,((1 \le i \le n \Longrightarrow x_i, \ldots, x_n \notin \mathrm{FV}(A_i))$.

LEMMA 9.3 (Thinning Lemma).

$$\Delta \subseteq \Delta'\ \&\ \Delta \vdash_{SA} M : A \Longrightarrow \Delta' \vdash_{SA} M : A.$$

PROOF. By induction on the derivation of $\Delta \vdash_{SA} M : A$. ⊣

LEMMA 9.4. *For a specification such that* (∗) *holds, if*

$$\Delta \vdash_{SA} M : A,$$

$\Delta =_\beta S(\Gamma)$ *and* Γ *an A-legal context, then there exist pseudoterms* M' *and* A' *such that* $M =_\beta M', A =_\beta A'$ *and*

$$\Gamma \vdash_A M' : A'.$$

Note that in fact A-legal means legal and, by Theorems 8.1 and 8.3, $\Gamma' \vdash_A M' : A'$ if and only if $\Gamma' \vdash M' : A'$.

PROOF. By induction on the derivation of $\Delta \vdash_{SA} M : A$ similar to the proof of Lemma 5.2. As we don't have the generalised *conversion rule* we need that Subject Reduction holds for the APTS, hence we had to assume that (∗) holds. In the case of the *abstraction rule* we use that condition (+) is strengthened as in Definition 9.1. ⊣

REMARK. In SAPTSs with non strengthened condition (+) the lemma need not hold. A counter-example is the following.

$$\mathscr{S} = \{s_1, s_2\},\ \mathscr{A} = \{s_1 : s_2\},\quad \mathscr{R} = \{(s_2, s_2, s_2), (s_2, s_1, s_2)\}.$$

This can be seen as follows. We have $\vdash_{SA} s_1 : s_2$ and $x : s_1 \vdash_{SA} s_1 : (\lambda y : s_1.y)s_2$ (by *generalised conversion*). Moreover the non strengthened condition (+) holds for s_2 and $(\lambda y : s_1.y)s_2$, hence we would get $\vdash_{SA} \lambda x : s_1.s_1 : \Pi x : s_1.(\lambda y : s_1.y)s_2$ and therefore $\vdash_{SA} \lambda x : s_1.s_1 : \Pi x : s_1.s_2$. But $\not\vdash \lambda x : s_1.s_1 : \Pi x : s_1.s_2$. Hence Lemma 9.4 would not hold. This counterexample holds because in SAPTSs you can replace $B \in \mathscr{C}$ by $C =_\beta B$, $C \notin \mathscr{C}$ by *generalised conversion.* Hence it is natural to strengthen (+) by: $B =_\beta B' \in \mathscr{C} \Longrightarrow (B' : s_2) \in \mathscr{A}$.

LEMMA 9.5. *If* Δ *is SA-legal for a given SAPTS such that* (∗) *holds, there is a context* Γ *that is A-legal for the APTS with the same specification, such that* $\Delta =_\beta S(\Gamma)$.

PROOF. As the proof of Lemma 5.3. ⊣

THEOREM 9.6 (Equivalence Theorem for APTSs and SAPTSs). *For any APTS and SAPTS with the same specification such that* $(*)$ *holds*

(i) $\Gamma \vdash_{\mathrm{A}} M : A \Longrightarrow S(\Gamma)$ *is SA-legal* & $S(\Gamma) \vdash_{\mathrm{SA}} M : A$.

(ii) Δ *is SA-legal* & $\Delta \vdash_{\mathrm{SA}} M : A \Longrightarrow (\exists \Gamma, M', A')\ [\Delta =_\beta S(\Gamma)$ & $M =_\beta M'$ & $A =_\beta A'$ & $\Gamma \vdash_{\mathrm{A}} M' : A']$.

PROOF. (ii) follows immediately from Lemmas 9.4 and 9.5.

(i) could be proved by induction on the derivation of $\Gamma \vdash_{\mathrm{A}} M : A$, but we give the proof by induction on the PTS-derivation of $\Gamma \vdash M : A$. ($(*)$ holds, hence the APTS is a PTS.) The proof is the same as the proof of Theorem 5.4, except for the *case of abstraction*:

$$\Gamma, x : B \vdash N : C \ \&\ \Gamma \vdash \Pi x : B.C : s_3$$

and $M \equiv \lambda x : B.N$, $A \equiv \Pi x : B.C$. By the Sharpened Generation Lemma for PTSs we get $\Gamma \vdash B : s_1$ by a shorter derivation than that of $\Gamma \vdash \Pi x : B.C : s_3$ & $\Gamma, x : B \vdash C : s_2$ & $(s_1, s_2, s_3) \in \mathscr{R}$. The induction hypothesis yields

$$S(\Gamma, x : B) \vdash_{\mathrm{SA}} N : C \ \&\ S(\Gamma) \vdash_{\mathrm{SA}} B : s_1.$$

Moreover the condition $(+)$ for SAPTSs holds for s_1 and C because of $\Gamma, x : B \vdash C : s_2$. Hence $\Gamma \vdash_{\mathrm{SA}} \lambda x : B.N : \Pi x : B.C$. ⊣

NOTE. 9.6 (i) does not hold in general for specifications that do not satisfy $(*)$. A counterexample is the following: Let as in Lemma 7.5 $\mathscr{S} = \{s_1, s_2, s_3\}$, $\mathscr{A} = \{s_1 : s_2, s_2 : s_3\}$ and $\mathscr{R} = \{(s_2, s_3, s_3), (s_3, s_3, s_3)\}$. Then we have in the APTS

$$\vdash_{\mathrm{A}} (\lambda y : s_2.s_1)s_1 : s_2 \ \&\ x : (\lambda y : s_2.s_1)s_1 \vdash_{\mathrm{A}} x : (\lambda y : s_2.s_1)s_1.$$

Moreover $(s_2, s_3, s_3) \in \mathscr{R}$ and $(\lambda y : s_2.s_1)s_1 \notin \mathscr{C}$. Hence

$$\vdash_{\mathrm{A}} \lambda x : ((\lambda y : s_2.s_1)s_1).x : \Pi x : ((\lambda y : s_2.s_1)s_1).((\lambda y : s_2.s_1)s_1).$$

In the corresponding SAPTS this does not hold because $(\lambda y : s_2.s_1)s_1 =_\beta s_1 \in \mathscr{C}$ and $s_1 : s_3 \notin \mathscr{A}$.

The following is a consequence of the preceding theorems.

LEMMA 9.7. *For any SPTS and SAPTS with the same specification such that* $(*)$ *holds*

(i) $\Delta \vdash_{\mathrm{S}} M : A$ & Δ *is S-legal* $\Longrightarrow \Delta \vdash_{\mathrm{SA}} M : A$.

(ii) $\Delta \vdash_{\mathrm{SA}} M : A$ & Δ *is SA-legal* $\Longrightarrow \Delta \vdash_{\mathrm{S}} M : A$.

PROOF.

(i) If Δ is S-legal and $\Delta \vdash_{\mathrm{S}} M : A$, then by Theorem 5.4 (ii) there are Γ, M' and A' such that $\Delta =_\beta S(\Gamma)$, $M =_\beta M'$, $A =_\beta A'$ and $\Gamma \vdash M' : A'$. By Theorem 8.1 then $\Gamma \vdash_{\mathrm{A}} M' : A'$ and by Theorem 9.6 (i) $S(\Gamma) \vdash_{\mathrm{SA}} M' : A'$. By the *conversion rule* $\Delta \vdash_{\mathrm{SA}} M : A$.

(ii) Similar using Theorems 9.6 (ii), 8.3 and 5.4 (i). ⊣

Note that as we can prove for any specification

$$(\Pi x : y.y) : s \vdash_{\mathrm{S}} (\lambda x : y.x) : (\Pi x : y.y)$$

but not the corresponding thing for $\vdash_{\mathrm{SA}}$, and for the specification of $\lambda 2$

$$y : \Box \vdash_{\mathrm{SA}} (\lambda x : y.x) : (\Pi x : y.y)$$

but not the corresponding thing for $\vdash_S$, it follows that neither part of Theorem 9.8 need hold if Δ is not S- (or SA-)legal.

THEOREM 9.8 (Equivalence Theorem for SPTSs and SAPTSs). *For any SPTS and SAPTS with the same specification such that* $(*)$ *holds*

$$\Delta \vdash_S M : A \ \&\ \Delta \text{ is S-legal} \iff \Delta \vdash_{SA} M : A \ \&\ \Delta \text{ is SA-legal.}$$

PROOF. This follows immediately from the preceding lemma by: If $\{x_1 : A_1, \ldots, x_n : A_n\}$ is S(A)-legal, then $\{x_1 : A_1, \ldots, x_i : A_i\}$ is S(A)-legal for each $i \leq n$. ⊣

§10. Summary of equivalences. We established a strong connection between PTSs, SPTSs, APTSs and SAPTSs. The following theorem is a compilation of Theorems 8.1, 8.3, 9.8 and 5.4.

THEOREM 10.1 (Equivalence Theorem for PTSs, APTSs, SPTSs and SAPTSs). *For any PTS, APTS, SPTS and SAPTS with the same specification such that* $(*)$ *holds*

(i) $\Gamma \vdash_A M : A \iff \Gamma \vdash M : A$.

(ii) $\Delta \vdash_{SA} M : A \ \&\ \Delta$ *is SA-legal* $\iff \Delta \vdash_S M : A \ \&\ \Delta$ *is S-legal* $\iff$ $(\exists \Gamma, M', A')\,[\Delta =_\beta S(\Gamma) \ \&\ M =_\beta M' \ \&\ A =_\beta A' \ \&\ \Gamma \vdash M' : A']$.

$(*)$ *is not needed for* (i) $\Leftarrow$ *and the second* $\Leftrightarrow$ *in* (ii) *and* $=_\beta$ *may be replaced by* $\twoheadrightarrow_\beta$ *in* (ii).

REFERENCES

[1] H. P. BARENDREGT, ***The lambda calculus, its syntax and semantics***, Studies in Logic and the Foundations of Mathematics, North-Holland, Amsterdam, 1984.

[2] ———, *Lambda calculi with types*, ***Handbook of logic in computer science, Volume II*** (S. Abramski, D. M. Gabbay, and T. S. E. Maibaum, editors), Oxford University Press, 1992.

[3] H. P. BARENDREGT, M.W . BUNDER, and W. J. M. DEKKERS, *Systems of illative combinatory logic complete for first-order propositional and predicate calculus*, this JOURNAL, vol. 58 (1993), pp. 769–788.

[4] L. S. VAN BENTHEM JUTTING, J. MCKINNA, and R. POLLACK, *Checking algorithms for pure type systems*, ***Types for proofs and programs***, Lecture Notes in Computer Science, vol. 806, 1994, pp. 19–61.

[5] S. BERARDI, *Towards a mathematical analysis of the Coquand-Huet calculus of constructions and the other systems of the Barendregt cube*, Department of Computer Science, Carnegie Mellon University and Dipartimento di Matematica, Universita di Torino, 1988.

[6] M. W. BUNDER, *Set theory based on combinatory logic*, ***Ph.D. thesis***, V. R. B. Groningen, 1969.

[7] ———, *Propositional and predicate calculus based on combinatory logic*, ***Notre Dame Journal of Formal Logic***, vol. 15 (1974), pp. 25–32.

[8] ———, *Scott's models and illative combinatory logic*, ***Notre Dame Journal of Formal Logic***, vol. 20 (1979), pp. 609–612.

[9] ———, *Conjunction without conditions in illative combinatory logic*, ***The Bulletin of the Section of Logic, Polish Academy of Sciences***, vol. 13 (1984), pp. 207–212.

[10] A. CHURCH, *A formulation of the simple theory of types*, this JOURNAL, vol. 5 (1940), pp. 56–68.

[11] T. COQUAND and G. HUET, *The calculus of constructions*, ***Information and Computation***, vol. 76 (1988), pp. 95–120.

[12] N. G. DE BRUIJN, *A survey of the AUTOMATH project*, ***To H. B. Curry: Essays on combinatory logic, lambda calculus and formalism*** (J. R. Hindley and J. P. Seldin, editors), Academic Press, 1980, pp. 580–606.

[13] W. J. M. DEKKERS, M. W. BUNDER, and H. P. BARENDREGT, *Completeness of the propositions-as-types interpretation of intuitionistic logic into illative combinatory logic*, this JOURNAL, vol. 63 (1998), no. 3, pp. 869–890.

[14] ———, *Completeness of two systems of illative combinatory logic for first-order propositional and predicate calculus*, ***Archive for Mathematical Logic***, vol. 37 (1998), pp. 327–341.

[15] J. H. GEUVERS, *Logics and type systems*, ***Ph.D. thesis***, Department of Computer Science, Catholic University, Toernooiveld 1, 6525 ED Nijmegen, The Netherlands, 1993.

[16] J.-Y. GIRARD, *Une extension de l'interpretation functionelle de Gödel à l'analyse et son application à l'élimination des coupures dans l'analyse et la théorie des types*, ***Proceedings of the second scandinavian logic symposium*** (J. E. Fenstad, editor), North-Holland, 1971.

[17] J. R. HINDLEY and J. P. SELDIN, ***Introduction to combinators and λ-calculus***, Cambridge University Press, 1986.

[18] F. KAMAREDDINE and R. NEDERPELT, *Canonical typing and* Π*-conversion in the Barendregt cube*, ***Journal of Functional Programming***, vol. 6 (1995), no. 2, pp. 245–267.

[19] Z. LUO, *ECC, an extended calculus of constructions*, ***Proceedings of the Fourth Annual Symposium on Logic in Computer Science, June 1989, Asilomar, California, USA***, 1989.

[20] MARTIN-LÖF, *An intuitionistic theory of types: predicative part*, ***Logic Colloquium '73***, North-Holland, 1982, pp. 153–175.

[21] E. POLL, *A typechecker for bijective pure type systems*, ***Computing Science Notes 93/22***, Department of Computer Science, University of Eindhoven, The Netherlands, 1993.

[22] R. POLLACK, *The theory of LEGO: A proof checker for the extended calculus of constructions*, ***Ph.D. thesis***, Department of Computer Science, University of Edinburgh, 1994.

[23] J. P. SELDIN, *On the proof theory of Coquand's calculus of constructions*, ***Annals of Pure and Applied Logic***, vol. 83 (1997), pp. 23–101.

[24] P. G. S. SEVERI, *Normalisation in lambda calculus and its relation to type inference*, ***Ph.D. thesis***, Department of Computer Science, University of Eindhoven, The Netherlands, 1996.

[25] J. TERLOUW, *Een nadere bewijstheoretische analyse van GSTT's*, Department of Computer Science, Catholic University, Toernooiveld 1, 6525 ED Nijmegen, The Netherlands, 1989.

FACULTY OF INFORMATICS
DEPARTMENT OF MATHEMATICS
UNIVERSITY OF WOLLONGONG
NSW 2522 AUSTRALIA
E-mail: martin_bunder@uow.edu.au

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
CATHOLIC UNIVERSITY
NIJMEGEN, THE NETHERLANDS
E-mail: wil@cs.kun.nl