The Subject Reduction Property in the $\lambda\Pi$-calculus modulo

Ronan Saillard
MINES ParisTech
ronan.saillard@cri.ensmp.fr

ABSTRACT

In type theory, the subject reduction (or type preservation) property states that the type of a λ-term is preserved under reduction. This article studies this property in the context of the $\lambda\Pi$-calculus modulo, a variant of the λ-calculus with dependent types ($\lambda\Pi$-calculus) where β-reduction is extended with user-defined object-level and type-level rewrite rules. We show that it is equivalent to the following property called Π-injectivity or product-compatibility: if product types are convertible then their components are pairwise convertible. We also show that subject reduction implies uniqueness of type and that both properties are undecidable. Finally we give a new decidable criterion ensuring subject reduction.

1. INTRODUCTION

In type theory, the subject reduction (or type preservation) property states that the type of a λ-term is preserved under reduction. Otherwise said, program execution (term reduction) is sound with respect to the static specification (type). This article studies this property in the context of the $\lambda\Pi$-calculus modulo [5], a variant of the λ-calculus with dependent types ($\lambda\Pi$-calculus) where β-reduction is extended with user-defined rewrite rules. Unlike in the $\lambda\Pi$-calculus, subject reduction does not hold in general in this setting, since the reduction might not be confluent. Moreover rewrite rules being explicitly added in the typing context, the rewriting system, hence subject reduction, now depends on the typing context.

An important motivation for this work comes from the development of *Dedukti* [4], a type checker for the $\lambda\Pi$-calculus modulo. Subject reduction is indeed used to prove the soundness of the classical inference algorithm for $\lambda\Pi$-calculus [10]. Thus a careful analysis of this property is needed to understand when this algorithm can be extended to the $\lambda\Pi$-calculus modulo. Moreover subject reduction is a crucial lemma in proving other properties of the reduction on typed terms such as confluence or termination.

After introducing the $\lambda\Pi$-calculus modulo in Section 2, we study, in Section 3, the property of uniqueness of type: two types of a single term are necessarily convertible. Then, in Section 4, we show that subject reduction is equivalent to Π-injectivity (also called product compatibility): a property on convertibility proofs between product types. In Section 5, we prove that both uniqueness of type and subject reduction are undecidable properties. This brings us, in Section 6, to look for decidable sufficient criteria to ensure these properties. The last two sections are devoted to proving the soundness of our main criterion.

2. THE $\lambda\Pi$-CALCULUS MODULO

In this section we describe the $\lambda\Pi$-calculus modulo, a variant of the λ-calculus with dependent types ($\lambda\Pi$-calculus) where β-reduction is extended with user-defined rewrite rules.

The syntax of the $\lambda\Pi$-calculus modulo for terms, rewrite rules and contexts is given in Figure 1. A term is either a variable, a constant, an application, a λ-abstraction, a product type or a sort (either **Type** or **Kind**). A pattern is a term made only of variables and constant applications (but it cannot be a variable). A rule context is a list of declarations, *i.e.*, pairs of a variable name and a term (its type). A rewrite rule is a triple made of a rule context, a pattern and a term. Finally a context is a list of declarations and rewrite rules.

x	$\in$	$\mathcal{V}$ (an infinite set)	(Variable)
f	$\in$	$\mathcal{F}$ (an infinite set)	(Constant)
s	$::=$	$\mathbf{Type} \mid \mathbf{Kind}$	(Sort)
t, u, A, B	$::=$	$x \mid f \mid t\ u \mid \lambda x^A.t \mid \Pi x^A.B \mid s$	(Term)
p	$::=$	$f\ \vec{p_0}$	(Pattern)
p_0	$::=$	$x \mid p$	
Δ	$::=$	$\emptyset \mid \Delta(x:t)$	(Rule Context)
r	$::=$	$[\Delta]\ p \hookrightarrow u$	(Rewrite Rule)
Γ	$::=$	$\emptyset \mid \Gamma(x:t) \mid \Gamma r$	(Context)

Figure 1: Syntax of the $\lambda\Pi$-calculus modulo.

As contexts may contain rewrite rules, they define a rewriting system. Given a context Γ we write $\rightarrow_\Gamma$ the smallest relation, compatible with the structure of terms, such that for any rule $[\Delta]\ l \hookrightarrow r$ in Γ and substitution σ with $dom(\sigma) = dom(\Delta)$, $\sigma l \rightarrow_\Gamma \sigma r$. We note $\rightarrow_\beta$ for β-reduction, $\rightarrow_{\beta^h}$ for head β-reduction, $\rightarrow_{\beta\Gamma}$ for $\rightarrow_\beta \cup \rightarrow_\Gamma$, $\rightarrow_{\Gamma^h}$ for head Γ-reduction and $\downarrow_{\beta\Gamma}$ for the joinability with respect to $\rightarrow_{\beta\Gamma}$.

The inference rules for the $\lambda\Pi$-calculus modulo, defining

$$\Gamma = \begin{cases} (T : \textbf{Type}) \\ (Bool : \textbf{Type}) \\ (Nat : \textbf{Type}) \\ (f : T) \\ (Z : Nat) \\ ([\emptyset]T \hookrightarrow \Pi x^{Nat}.Nat) \\ ([\emptyset]T \hookrightarrow \Pi x^{Nat}.Bool) \end{cases} \qquad \begin{array}{l} \Gamma \vdash f\ Z : Nat \\ \Gamma \vdash f\ Z : Bool \\ \text{But} \\ Nat \not\mathbb{W}_\Gamma Bool \end{array}$$

Figure 3: Uniqueness of Type: A Counterexample

well-formed contexts and well-typed terms, are given in Figure 2. One can notice that these rules differ from the rules of λΠ-calculus only in two points. First, the relation in the **Conv** rule is extended from $\downarrow_\beta$ to $\downarrow_{\beta\Gamma}$, allowing for more terms to be typed. Secondly, there is a new rule **Rw** which makes possible the addition of rewrite rules in the context, thus extending explicitly the considered rewriting system. This latter rule is a novelty with respect to previous formalizations (See [8] for a discussion on this rule). Addition of rewrite rules becomes an iterative process: rules previously added can be used to type new rules.

We write $T_1 \ \mathbb{C}_\Gamma \ T_2$ when $T_1 \downarrow_{\beta\Gamma} T_2$ and both T_1 and T_2 are typable in Γ. We note $\mathbb{W}_\Gamma$ the reflexive and transitive closure of $\mathbb{C}_\Gamma$.

3. UNIQUENESS OF TYPE

We first look at the property of uniqueness of type: a term has one type up to convertibility. This property does not hold in general in the λΠ-calculus modulo as exemplified in Figure 3.

The purpose of this section is twofold: first to give an alternative formulation of uniqueness of type so that it becomes clear, in the next section, that subject reduction entails uniqueness of type, secondly to prove important lemmas used in the rest of this paper.

THEOREM 3.1. *The following properties are equivalent:*

- *(Uniqueness of Type) if* $\Gamma \vdash t : T_1$ *and* $\Gamma \vdash t : T_2$ *then* $T_1 = T_2 = \textbf{Kind}$ *or* $T_1 \ \mathbb{W}_\Gamma \ T_2$.
- *(Right-Π-Injectivity) if* $\Pi x^A.B_1 \ \mathbb{W}_\Gamma \ \Pi x^A.B_2$ *then* $B_1 \ \mathbb{W}_{\Gamma(x:A)} \ B_2$.

Before proving this theorem we need to prove a technical lemma, the so-called inversion lemma.

LEMMA 3.2 (INVERSION). *Assume that* $\Gamma \vdash t : T$.

- *If* $t = \textbf{Type}$ *then* $T = \textbf{Kind}$.
- *If* $t = x$ *then* $\Gamma(x) \ \mathbb{W}_\Gamma \ T$.
- *If* $t = fu$ *then there exist* A *and* B *such that* $\Gamma \vdash f : \Pi x^A.B$, $\Gamma \vdash u : A$ *and* $B[x/u] \ \mathbb{W}_\Gamma \ T$.
- *If* $t = \lambda x^A.t$ *then there exists* B *such that* $\Gamma \vdash A : \textbf{Type}$, $\Gamma(x : A) \vdash t : B$, $B \neq \textbf{Kind}$ *and* $\Pi x^A.B \ \mathbb{W}_\Gamma \ T$.
- *If* $t = \Pi x^A.B$ *then* $\Gamma \vdash A : \textbf{Type}$ *and* $\Gamma(x : A) \vdash B : T$ *and* $T = \textbf{Kind}$ *or* $T = \textbf{Type}$.

PROOF. By induction on the typing derivation. □

PROOF OF THEOREM 3.1. First we prove uniqueness of type under the hypothesis of right-Π-injectivity. We proceed by induction on the first typing derivation. We only detail the **(App)** case since it is only case which differs from the standard proof (for λΠ-calculus).

- **(App)** We have $t = uv$, $T_1 = B_1[x/v]$, $\Gamma \vdash u : \Pi x^{A_1}.B_1$ and $\Gamma \vdash v : A_1$. By inversion of t, there exist A_2, B_2 such that $\Gamma \vdash u : \Pi x^{A_2}.B_2$, $\Gamma \vdash v : A_2$ and $T_2 = B_2[x/v]$. By induction hypothesis we have $\Pi x^{A_1}.B_1 \ \mathbb{W}_\Gamma \ \Pi x^{A_2}.B_2$ and $A_1 \ \mathbb{W}_\Gamma \ A_2$. Thus $\Pi x^{A_2}.B_1 \ \mathbb{W}_\Gamma \ \Pi x^{A_1}.B_1 \ \mathbb{W}_\Gamma \ \Pi x^{A_2}.B_2$ and by right-Π-injectivity $B_1 \ \mathbb{W}_{\Gamma(x:A_2)} \ B_2$. It follows that $T_1 = B_1[x/v] \ \mathbb{W}_\Gamma \ B_2[x/v] = T_2$.

Now we prove right-Π-injectivity using uniqueness of type. Assume that $\Pi x^A.B_1 \ \mathbb{W}_\Gamma \ \Pi x^A.B_2$, then we can derive $\Gamma(f : \Pi x^A.B_1)(x : A) \vdash fx : B_1$ and $\Gamma(f : \Pi x^A.B_1)(x : A) \vdash fx : B_2$. By uniqueness of type we have $B_1 \ \mathbb{W}_{\Gamma(x:A)} \ B_2$. □

4. SUBJECT REDUCTION

As for uniqueness of type, the subject reduction does not hold in general (Figure 4). Subject reduction is usually proved using a property called Π-injectivity (or product-compatibility). We show hereafter that the two properties are actually equivalent. Since Π-injectivity obviously implies right-Π-injectivity, we get as a corollary that subject reduction entails uniqueness of type.

THEOREM 4.1. *The following properties are equivalent:*

- *(Subject Reduction) if* $\Gamma \vdash t : T$ *and* $t \to_{\beta\Gamma} t'$ *then* $\Gamma \vdash t' : T$.
- *(Π-Injectivity) if* $\Pi x^{A_1}.B_1 \ \mathbb{W}_\Gamma \ \Pi x^{A_2}.B_2$ *then* $A_1 \ \mathbb{W}_\Gamma \ A_2$ *and* $B_1 \ \mathbb{W}_{\Gamma(x:A_2)} \ B_2$.

$$\Gamma = \begin{cases} (T : \textbf{Type}) \\ (Bool : \textbf{Type}) \\ (false : Bool) \\ (Nat : \textbf{Type}) \\ (Z : Nat) \\ ([\emptyset]T \hookrightarrow \Pi x^{Nat}.Nat) \\ ([\emptyset]T \hookrightarrow \Pi x^{Bool}.Bool) \end{cases} \qquad \begin{array}{l} \text{We have} \\ \Pi x^{Bool}.Bool \ \mathbb{W}_\Gamma \ \Pi x^{Nat}.Nat \\ \text{thus} \\ \Gamma \vdash \lambda x^{Nat}.Z : \Pi x^{Bool}.Bool \\ \text{and} \\ \Gamma \vdash (\lambda x^{Nat}.Z) false : Bool \\ \text{But} \\ \Gamma \nvdash Z : Bool \end{array}$$

Figure 4: Subject Reduction: A Counterexample.

COROLLARY 4.2. *Subject reduction implies uniqueness of type.*

The proof of the theorem uses the following lemma.

LEMMA 4.3 (HEAD SUBJECT REDUCTION). *If* $\Gamma \vdash t : T$, $t \to^h_\Gamma t'$ *and* Π*-injectivity holds for* Γ *then* $\Gamma \vdash t' : T$.

PROOF. We will use the following lemma: for every pattern p if we have $\Gamma_0\Delta \vdash p : T_0$ and $\Gamma_0\Gamma_1 \vdash \sigma p : T$ then for all $x \in \Delta \cap FV(p)$ we have $\Gamma_0\Gamma_1 \vdash \sigma x : \sigma(\Delta(x))$. This can be proved by induction on p. It requires Π-injectivity.

$$\textbf{(Empty)}\ \frac{}{\emptyset\ wf} \qquad \textbf{(Dec)}\ \frac{\Gamma\ wf \qquad \Gamma \vdash A : s \qquad f \notin \Gamma}{\Gamma(f : A)\ wf}$$

$$\textbf{(Rw)}\ \frac{\Gamma\ wf \qquad \Gamma\Delta \vdash l : T \qquad \Gamma\Delta \vdash r : T \qquad FV(r) \cap \Delta \subset FV(l) \qquad l \text{ is a pattern}}{\Gamma([\Delta]l \hookrightarrow r)\ \text{wf}}$$

$$\textbf{(Type)}\ \frac{\Gamma\ wf}{\Gamma \vdash \textbf{Type} : \textbf{Kind}} \qquad \textbf{(Var/Cst)}\ \frac{\Gamma\ wf \qquad (x : A) \in \Gamma \qquad x \in \mathcal{V} \cup \mathcal{F}}{\Gamma \vdash x : A}$$

$$\textbf{(App)}\ \frac{\Gamma \vdash t : \Pi x^A.B \qquad \Gamma \vdash u : A}{\Gamma \vdash tu : B[x\backslash u]} \qquad \textbf{(Conv)}\ \frac{\Gamma \vdash t : A \qquad \Gamma \vdash B : s \qquad A \downarrow_{\beta\Gamma} B}{\Gamma \vdash t : B}$$

$$\textbf{(Abs)}\ \frac{\Gamma \vdash A : \textbf{Type} \qquad \Gamma(x : A) \vdash t : B \qquad B \neq \textbf{Kind}}{\Gamma \vdash \lambda x^A.t : \Pi x^A.B}$$

$$\textbf{(Prod)}\ \frac{\Gamma \vdash A : \textbf{Type} \qquad \Gamma(x : A) \vdash B : s}{\Gamma \vdash \Pi x^A.B : s}$$

Figure 2: Typing rules for $\lambda\Pi$-calculus modulo.

Now assume $\Gamma \vdash t : T$ and $t \rightarrow^h_\Gamma t'$ then $\Gamma = \Gamma_0([\Delta]l \hookrightarrow r)\Gamma_1$, $t = \sigma l$, $t' = \sigma r$, $\Gamma_0\Delta \vdash l : T_0$ and $\Gamma_0\Delta \vdash r : T_0$. From the lemma above, it follows that $\Gamma \vdash \sigma l : \sigma T_0$ and $\Gamma \vdash \sigma r : \sigma T_0$. By uniqueness of type we have $T \mathbb{W}_\Gamma \sigma T_0$. It follows that $\Gamma \vdash \sigma r : T$. □

We now prove the first half of the theorem.

Proof of theorem 4.1 ($\Leftarrow$). We proceed by induction on t. The interesting case is when t is a redex. When t is a Γ-redex then we can apply lemma 4.3. Now assume that t is a β-redex. We have $t = (\lambda x^{A_1}.u_0)v$ and $t' = u_0[x/v]$. Then, by inversion, there exist A_2, B_1, B_2 such that $\Gamma(x : A_1) \vdash u_0 : B_1$, $\Gamma \vdash v : A_2$, $B_2[x/v] \mathbb{W}_\Gamma T$ and $\Pi x^{A_1}.B_1 \mathbb{W}_\Gamma \Pi x^{A_2}.B_2$. By Π-injectivity, $A_1 \mathbb{W}_\Gamma A_2$ and $B_1 \mathbb{W}_{\Gamma(x:A_2)} B_2$. Therefore we have $\Gamma \vdash v : A_1$ and $\Gamma \vdash u_0[x/v] : B_1[x/v]$. Finally we have $B_1[x/v] \mathbb{W}_\Gamma B_2[x/v] \mathbb{W}_\Gamma T$ and $\Gamma \vdash u_0[x/v] : T$. □

Some more lemmas are needed to complete the proof of the other implication.

A rewrite rule is said to be left-linear if each variable in its left-hand side appears exactly once. A rewriting system is said to be left-linear if all its rewrite rules are left-linear.

Lemma 4.4. *If $\rightarrow_\Gamma$ is left-linear and confluent then the subject reduction property holds.*

Proof. Since $\rightarrow_\Gamma$ is left-linear and confluent, $\rightarrow_{\beta\Gamma}$ is confluent [11]. Then if $\Pi x^{A_1}.B_1 \mathbb{W}_\Gamma \Pi x^{A_2}.B_2$, we have $\Pi x^{A_1}.B_1 \downarrow \Pi x^{A_2}.B_2$. Thus we also have $A_1 \downarrow A_2$ and $B_1 \downarrow B_2$. It follows that $A_1 \mathbb{W}_\Gamma A_2$ and $B_1 \mathbb{W}_{\Gamma(x:A_2)} B_2$. □

Lemma 4.5 (Non-dependent Π-injectivity). *If $\Pi x^{A_1}.B_1 \mathbb{W}_\Gamma \Pi x^{A_2}.B_2$, $\Gamma \vdash a : A_2$, B_1 does not depend on x and subject reduction holds for Γ then $B_1 \mathbb{W}_\Gamma B_2[x/a]$.*

Proof. From $\Pi x^{A_1}.B_1 \mathbb{W}_\Gamma \Pi x^{A_2}.B_2$ and $\Gamma \vdash a : A_2$ we can derive $\Gamma(b_1 : B_1) \vdash (\lambda x^{A_1}.b_1)a : B_2[x/a]$. By subject reduction $\Gamma(b_1 : B_1) \vdash b_1 : B_2[x/a]$. Finally, by inversion $B_1 \mathbb{W}_\Gamma B_2[x/a]$. □

Lemma 4.6 (Identity Trick). *If $\Gamma \vdash (\lambda x^A.x)a : T$ and subject reduction holds for Γ then $A \mathbb{W}_\Gamma T$.*

Proof. This follows from inversion and non-dependent Π-injectivity. □

Now we can prove that subject reduction implies Π-injectivity.

Proof of theorem 4.1 ($\Rightarrow$). Assume $\Pi x^{A_1}.B_1 \mathbb{W}_\Gamma \Pi x^{A_2}.B_2$.

- From $\Gamma(a_2 : A_2)(f : \Pi x^{A_1}.B_1) \vdash (\lambda x^{A_1}.(f((\lambda y^{A_1}.y)x)))a_2 : B_2[x/a_2]$, we can deduce by subject reduction $\Gamma(a_2 : A_2)(f : \Pi x^{A_1}.B_1) \vdash f((\lambda y^{A_1}.y)a_2) : B_2[x/a_2]$. The set of typed terms being closed by taking a subterm, we have for some T, $\Gamma(a_2 : A_2)(f : \Pi x^{A_1}.B_1) \vdash (\lambda y^{A_1}.y)a_2 : T$. Then by the identity trick we have $A_1 \mathbb{W}_\Gamma T$. Reducing further we have $\Gamma(a_2 : A_2)(f : \Pi x^{A_1}.B_1) \vdash a_2 : T$. Thus by inversion $T \mathbb{W}_\Gamma A_2$. It follows that $A_1 \mathbb{W}_\Gamma A_2$.
- From $\Gamma(x : A_2)(f : \Pi x^{A_1}.B_1) \vdash (\lambda y^{A_1}.((\lambda z^{B_1[x/y]}.z)(fy)))x : B_2$, we can deduce by subject reduction $\Gamma(x : A_2)(f : \Pi x^{A_1}.B_1) \vdash (\lambda z^{B_1}.z)(fx) : B_2$. By the identity trick we have $B_1 \mathbb{W}_{\Gamma(x:A_2)} B_2$.

□

5. UNDECIDABILITY

In this section, we prove that both uniqueness of type and subject reduction are undecidable properties. We follow a strategy similar to the one used to prove undecidability of confluence in [9].

Lemma 5.1. *Right-Π-injectivity and Π-injectivity are undecidable properties.*

Corollary 5.2. *Uniqueness of type and subject reduction are undecidable properties.*

Proof. Follows from theorem 3.1, 4.1 and 5.1. □

$x(yz)$	$=$	$(xy)z$
$abaabb$	$=$	$bbaaba$
$aababba$	$=$	$bbaaaba$
$abaaabb$	$=$	$abbabaa$
$bbbaabbaaba$	$=$	$bbbaabbaaaa$
$aaaabbaaba$	$=$	$bbaaaa$

Figure 5: Equational theory with undecidable word problem.

We will use the following lemma due to Matijasevich [6].

Lemma 5.3. *The word problem for the set of equations E of Table 5 is undecidable.*

Proof of lemma 5.1. We reduce the word problem for E to (right-)Π-injectivity. We will consider the following typing context, where we introduce a type w for words, a term $\dot{\epsilon}$ representing the empty word, and two symbols $\dot{a}$ and $\dot{b}$ corresponding to the letters a and b:

$$\Gamma_0 = (w : \mathbf{Type})(\dot{\epsilon} : w)(\dot{a} : \Pi x^w.w)(\dot{b} : \Pi x^w.w)$$

and the following translation $|.|(x)$ from words to terms of type w (with the term x as parameter):

$$|a|(x) \mapsto \dot{a}x, \quad |b|(x) \mapsto \dot{b}x, \quad |m_1.m_2|(x) \mapsto |m_1|(|m_2|(x)),$$

Let Γ be the context obtained by adding to Γ_0, for each equation $w_1 = w_2$ in E, the rewrite rules $[(x : w)]|w_1|(x) \hookrightarrow |w_2|(x)$ and $[(x : w)]|w_2|(x) \hookrightarrow |w_1|(x)$.

Note that $\rightarrow_{\beta\Gamma}$ is confluent since $\rightarrow_{\Gamma}$ is left-linear and confluent. Thus we have $m_1 =_E m_2$ if and only if $|m_1|(\dot{\epsilon}) \downarrow_{\Gamma} |m_2|(\dot{\epsilon})$ if and only if $|m_1|(\dot{\epsilon}) \downarrow_{\beta\Gamma} |m_2|(\dot{\epsilon})$.

Now let m_1 and m_2 be arbitrary words and consider the following context:

$$\begin{aligned}\Gamma_2 = \quad & \Gamma(T : \mathbf{Type})(A : \mathbf{Type})(B : \Pi x^w.\mathbf{Type}) \\ & ([\emptyset]T \hookrightarrow \Pi x^A.B(|m_1|(\dot{\epsilon}))) \\ & ([\emptyset]T \hookrightarrow \Pi x^A.B(|m_2|(\dot{\epsilon})))\end{aligned}$$

We now prove that $m_1 =_E m_2$ if and only if (right-)Π-injectivity holds for $\Gamma_{(m_1,m_2)}$.

- Assume that $m_1 =_E m_2$. Since $\rightarrow_{\Gamma_2}$ is left-linear and confluent, $\rightarrow_{\beta\Gamma_2}$ is confluent. Hence (right-)Π-injectivity holds.
- Assume (right-)Π-injectivity.
  Then since $\Pi x^A.B|m_1| \, \mathbb{W}_{\Gamma_2} \, \Pi x^A.B|m_2|$
  we have $B|m_1| \, \mathbb{W}_{\Gamma_2(x:A)} \, B|m_2|$.
  It follows that $|m_1| \, \mathbb{W}_{\Gamma_2(x:A)} \, |m_2|$ and $m_1 =_E m_2$.

□

6. CRITERIA FOR SUBJECT REDUCTION

We have seen in the previous section that subject reduction is an undecidable property. We now try to find decidable criteria that ensure this property.

We are motivated by the fact that we want to extend the standard type-checking algorithm for $\lambda\Pi$-calculus to the whole $\lambda\Pi$-calculus modulo. Indeed the soundness of the algorithm relies on the subject reduction property. Thus being able to automatically determine if subject reduction holds permits to automatically detect when the type checking algorithm can be safely used.

We have already given a first criterion in lemma 4.4.

Criterion 1. *$\rightarrow_{\Gamma}$ is left-linear and confluent.*

Of course the confluence of $\rightarrow_{\Gamma}$ is not decidable, but there exist numerous criteria for proving confluence of left-linear term rewriting system that can be used here. Moreover tools automating this task exist. See for instance the competitors of the Confluence Competition [1].

We now give a purely syntactical criterion. It is a refinement of a criterion due to Barbanera *et al.* [2] and later extended by Blanqui [3].

Criterion 2. *No product types appear in the right-hand side of rewrite rules.*

Theorem 6.1 (Main Theorem). *If the rewrite rules in Γ verify Criterion 2 then subject reduction holds for Γ.*

Section 8 is devoted to the proof of this theorem.

As a corollary we get that subject reduction holds in the $\lambda\Pi$-calculus modulo restricted to object-level rewrite rules.

Criterion 1 is actually a criterion for the confluence of $\rightarrow_{\beta\Gamma}$. Another convenient way to prove the confluence of a rewriting system is to use Newman's lemma [7]: a locally confluent terminating rewriting system is confluent. So one could think that Newman's lemma could serve as a basis for another criterion for subject reduction. However this seems not to be a good solution in practice. Indeed proving the termination of $\rightarrow_{\beta\Gamma}$ without confluence nor subject reduction is a rather difficult task and it seems that, as soon as we consider type-level rewrite rules, proofs of termination in $\lambda\Pi$-calculus modulo need confluence [3].

7. WEAK TYPING

Before proving the soundness of the criterion in Section 8, we need to introduce a technical tool: weak typing. The idea is to type the terms of the $\lambda\Pi$-calculus modulo as in the simply typed λ-calculus. All typed terms are also weakly-typed and, in this setting, we can prove a weak version of the subject reduction property. As a corollary we get that the reducts of a typed term are weakly typed and this can be used to prove that they respect certain properties.

The syntax of simple types and simple contexts is in given in Figure 7.

$$\begin{array}{lll l}
F & \in & \mathcal{V} \cup \mathcal{F} & \text{(Atomic Type)} \\
A, B & ::= & F \mid A \to B \mid \textbf{Type} \mid \textbf{Kind} & \text{(Simple Type)} \\
\Gamma & ::= & \emptyset \mid \Gamma(x : A) \mid \Gamma(F \hookrightarrow B) & \text{(Simple Context)}
\end{array}$$

Figure 6: Syntax of Simple Types

We define Figure 7 a translation from terms in $\lambda\Pi$-calculus modulo to simple types and from contexts to simple contexts.

$$\begin{array}{ll}
||\textbf{Kind}|| = \textbf{Kind} & ||x|| = x \\
||\textbf{Type}|| = \textbf{Type} & ||f|| = f \\
||tu|| = ||t|| & ||\lambda x^A.t|| = ||t|| \\
||\Pi x^A.B|| = ||A|| \to ||B|| & \\
 & \\
||\Gamma(x : A)|| = ||\Gamma||(x : ||A||) & ||\emptyset|| = \emptyset \\
||\Gamma([\Delta]l \hookrightarrow r)|| = ||\Gamma||(||l|| \hookrightarrow ||r||) &
\end{array}$$

Figure 7: Translation from λ-terms to Simple Types.

As for contexts, simple contexts give rise to a rewriting system on simple types. We will also use the notations $\to_\Gamma$ and $\downarrow_\Gamma$ for this system.

We give, in Figure 8, the typing rules for the weak $\lambda\Pi$-calculus modulo.

Lemma 7.1. *If $A \downarrow_{\beta\Gamma} B$ then $||A|| \downarrow_{||\Gamma||} ||B||$.*

Proof. It is easy to see that if $t \to_{\beta\Gamma} t'$ then either $||t|| = ||t'||$ or $||t|| \to_{||\Gamma||} ||t'||$. □

Lemma 7.2 (Soundness of the translation). *If $\Gamma \vdash t : T$ then $||\Gamma|| \vdash_w t : ||T||$*

Proof. Induction on the typing derivation. We detail the **(Conv)** case:

- **(Conv)** We have $\Gamma \vdash t : A$, $\Gamma \vdash B : s$ and $A \downarrow_{\beta\Gamma} B$. By induction hypothesis we have $||\Gamma|| \vdash_w t : ||A||$ and $||\Gamma|| \vdash B : ||s||$. Moreover by lemma 7.1, we have $||A|| \downarrow_{||\Gamma||} ||B||$. Thus we can use the **(Conv$_w$)** to deduce $||\Gamma|| \vdash_w t : ||B||$.

□

Now let us assume that no rewrite rules have an arrow (→) in its right-hand side. We have the following lemma:

Lemma 7.3 (Weak Π-injectivity). *If $A_1 \to B_1 \mathbb{W}^w_\Gamma A_2 \to B_2$ then $A_1 \mathbb{W}^w_\Gamma A_2$ and $B_1 \mathbb{W}^w_\Gamma B_2$.*

We can now prove a version of subject reduction for this system.

Lemma 7.4 (Weak Subject Reduction). *If $\Gamma \vdash_w t : T$ and $t \to_{\beta\Gamma} t'$ then $\Gamma \vdash_w t' : T$*

Proof. The proof is the same as for lemma 4.1 but we use weak Π-injectivity (lemma 7.3) instead of Π-injectivity. □

Lemma 7.5. *If $\Gamma \vdash_w tu : T$ then u cannot contain a product type.*

Proof. This is a corollary of the following properties:

- A term containing a product-type can only have a sort as its type.
- No term has type **Type** → B.

Each of these statements can be proved by induction on the typing derivation. □

8. PROOF OF THE MAIN THEOREM

Now the proof is essentially the one that can be found in [2] and [3]. The main difference lies in the postponement lemma (Lemma 8.2) where we make use of weak typing.

Lemma 8.1 (Commutation). *If $t \to^*_{\beta h} u$ and $t \to^*_\Gamma v$ then there exists t' such that $u \to^*_\Gamma t'$ and $v \to^*_{\beta h} t'$.*

Proof. Induction on the number of h-steps. u_0 is a β-redex so w is also a β-redex and assuming that $u_0 = (\lambda x^A.f)a$ we have $u = f[x/a]$ and $w = (\lambda x^{A'}.f')a'$ with $A \to^*_\Gamma A'$, $f \to^*_\Gamma f'$ and $a \to^*_\Gamma a'$. We take $t' = f'[x/a']$.

$$\begin{tikzcd}
t \arrow[r, "h^n"'] \arrow[d, "\Gamma^*"'] & u_0 \arrow[r, "h^1"'] \arrow[d, dotted, "\Gamma*"'] \arrow[dl, phantom, "(IH)"] & u \arrow[d, dotted, "\Gamma^*"'] \\
v \arrow[r, dotted, "h^*"'] & w \arrow[r, dotted, "h^1"'] & t'
\end{tikzcd}$$

□

Lemma 8.2 (Postponement). *Assume that right-hand sides of rewrite rules do not contain product types. If $\Gamma \vdash t :$ **Type** and $t \to^*_\Gamma t' \to^*_{\beta h} \Pi x^{A_1}.B_1$ then there exist A_2, B_2 such that $t \to^*_{\beta h} \Pi x^{A_2}.B_2$, $A_2 \to^*_\Gamma A_1$ and $B_2 \to^*_\Gamma B_1$.*

Proof. First we remark that $t' = (\lambda \vec{x}^{\vec{A}}.\Pi x^{A'_1}.B'_1)\vec{u}$ with $A_1 = A'_1[\vec{x}/\vec{u}]$ and $B_1 = B'_1[\vec{x}/\vec{u}]$. Indeed the only other possible form for t' is $(\lambda \vec{x}^{\vec{A}}.x_i\vec{v})\vec{u}$ but this assumes that there is a product type in $\vec{u}$ or $\vec{v}$, which is impossible by Lemma 7.5. For similar reasons, Lemma 7.5 also prevents Γ-reduction from introducing product types. Thus $t = (\lambda \vec{y}^{\vec{A'}}.\Pi x^{A'_2}.B'_2)\vec{u'}$ with $A'_2 \to^*_\Gamma A'_1$, $B'_2 \to^*_\Gamma B'_1$ and $\vec{u'} \to^*_\Gamma \vec{u}$. We choose $A_2 = A'_2[\vec{y}/\vec{u'}]$ and $B_2 = B'_2[\vec{y}/\vec{u'}]$. □

Lemma 8.3 (Π-injectivity for Kinds). *If $\Pi x^{A_1}.B_1 \mathbb{W}_\Gamma \Pi x^{A_2}.B_2$ with $\Gamma \vdash \Pi x^{A_1}.B_1 :$ **Kind** then $A_1 \mathbb{W}_\Gamma A_2$ and $B_1 \mathbb{W}_{\Gamma(x:A_2)} B_2$.*

$$(\textbf{Empty}_w)\ \frac{}{\emptyset\ wf_w} \qquad (\textbf{Dec}_w)\ \frac{\Gamma\ wf_w \quad \Gamma \vdash_w A : s \quad f \notin \Gamma}{\Gamma(f : A)\ wf_w}$$

$$(\textbf{Rw}_w)\ \frac{\Gamma\ wf_w \quad \Gamma\Delta \vdash_w l : T \quad \Gamma \vdash_w r : T}{\Gamma(l \hookrightarrow r)\ wf_w}$$

$$(\textbf{Type}_w)\ \frac{\Gamma\ wf_w}{\Gamma \vdash_w \textbf{Type} : \textbf{Kind}} \qquad (\textbf{Var/Cst}_w)\ \frac{\Gamma\ wf_w \quad (x : A) \in \Gamma \quad x \in \mathcal{V} \cup \mathcal{F}}{\Gamma \vdash_w x : A}$$

$$(\textbf{App}_w)\ \frac{\Gamma \vdash_w t : A \to B \quad \Gamma \vdash_w u : A}{\Gamma \vdash_w tu : B} \qquad (\textbf{Conv}_w)\ \frac{\Gamma \vdash_w t : A \quad \Gamma \vdash_w B : s \quad A \downarrow_{\beta\Gamma} \|B\|}{\Gamma \vdash_w t : \|B\|}$$

$$(\textbf{Abs}_w)\ \frac{\Gamma \vdash_w A : \textbf{Type} \quad \Gamma(x : \|A\|) \vdash_w t : B \quad B \neq \textbf{Kind}}{\Gamma \vdash_w \lambda x^A.t : \|A\| \to B}$$

$$(\textbf{Prod}_w)\ \frac{\Gamma \vdash_w A : \textbf{Type} \quad \Gamma(x : \|A\|) \vdash_w B : s}{\Gamma \vdash_w \Pi x^A.B : s}$$

Figure 8: Typing rules for weak λΠ-calculus modulo.

PROOF. $\Pi x^{A_2}.B_2$ and all the intermediate types are of type **Kind** so they are product types. □

LEMMA 8.4 (SUBJECT REDUCTION FOR β^h ON TYPES). *If* $\Gamma \vdash t : \textbf{Type}$ *and* $t \to_{\beta^h} t'$ *then* $\Gamma \vdash t' : \textbf{Type}$.

PROOF. Induction on the typing derivation:

- **(App)** Same as in the proof of the 4.1 but using lemma 8.3 instead of full Π-injectivity.
- **(Conv)** $B = \textbf{Type}$ and $A = \textbf{Type}$ so, by (IH), $\Gamma \vdash t' : \textbf{Type}$.
- No other inference rule can apply.

□

LEMMA 8.5. *Let* λΠ *be the rewrite rules of* λΠ*-calculus modulo (Figure 2). Now let* λΠ′ *be the same rules but replacing, in the* **(Conv)** *rule, the joinability relation* $\downarrow_{\beta\Gamma}$ *by* $\downarrow_\beta \cup \downarrow_\Gamma$. *If subject reduction holds for a context* Γ *in* λΠ′ *then subject reduction holds for* Γ *in* λΠ.

PROOF. Let $\vdash'$ be the typing relation for λΠ′. We show that $\vdash\ =\ \vdash'$. We obviously have $\vdash\ \supset\ \vdash'$. We show the converse inclusion by induction on $\vdash$. The only non-trivial case is of course the **(Conv)** rule.

- **(Conv)** We have $\Gamma \vdash t : A$, $\Gamma \vdash B : s$ and $A \downarrow_{\beta\Gamma} B$. By induction hypothesis we also have $\Gamma \vdash' t : A$ and $\Gamma \vdash' B : s$. We can decompose the conversion in
$$A \to_\beta A_1 \to_\Gamma A_2 \to_\beta \ldots \leftarrow_\beta B_2 \leftarrow_\Gamma B_1 \leftarrow_\beta B$$
but A and B are typable for $\vdash'$ so, by subject reduction, all the A_i and B_i are typable. Thus we can replace the conversion in λΠ by a sequence of conversions in λΠ′.

□

PROOF OF THE MAIN THEOREM. Following lemma 8.5 we will replace the joinability relation $\downarrow_{\beta\Gamma}$ by $\downarrow_\beta \cup \downarrow_\Gamma$.

Now we proceed by induction on the number of valleys to prove the following property: if $\Pi x^{A_1}.B_1 \mathbb{W}_\Gamma T$ then $T \to_{\beta^h} \Pi x^{A_2}.B_2$ with $A_1 \mathbb{W}_\Gamma A_2$ and $B_1 \mathbb{W}_{\Gamma(x:A_2)} B_2$. This amounts to proving for $\downarrow = \downarrow_\beta$ and $\downarrow = \downarrow_\Gamma$ that if $T_1 \downarrow T_2$, $T_1 \to^*_{\beta^h} \Pi x^{A_1}.B_1$ and $\Gamma \vdash T_i : \textbf{Type}$ then there exist A_2, B_2 such that $T_2 \to_{\beta^h} \Pi x^{A_2}.B_2$, $A_1 \mathbb{W}_\Gamma A_2$ and $B_1 \mathbb{W}_{\Gamma(x:A_2)} B_2$.

- If $\downarrow = \downarrow_\beta$ then it follows from confluence of $\to_\beta$ and standardization.

$$\begin{tikzcd}
T_1 \arrow[d, "h^*"'] \arrow[dr, "\beta^*"] & & T_2 \arrow[dl, "\beta^*"'] \arrow[d, dotted, "h^*"] \\
\Pi x^{A_1}.B_1 \arrow[dr, dotted, "\beta^*"'] & U \arrow[d, dotted, "\beta^*"] & \Pi x^{A_2}.B_2 \arrow[dl, dotted, "\beta^*"] \\
& \Pi x^{A_0}.B_0 &
\end{tikzcd}$$

- If $\downarrow = \downarrow_\Gamma$ then it follows from commutation 8.1 and postponement 8.2.

$$\begin{tikzcd}
T_1 \arrow[d, "h^*"'] \arrow[dr, "\Gamma^*"] & & T_2 \arrow[dl, "\Gamma^*"'] \arrow[d, dotted, "h^*"'] \\
\Pi x^{A_1}.B_1 \arrow[dr, dotted, "\Gamma^*"'] & U \arrow[d, dotted, "h^*"] & \Pi x^{A_2}.B_2 \arrow[dl, dotted, "\Gamma^*"] \\
& \Pi x^{A_0}.B_0 &
\end{tikzcd}$$

All the A_i and B_i are well-typed by subject reduction for $\to_{\beta^h}$. □

9. CONCLUSION

We have studied the subject reduction property in the context of the λΠ-calculus modulo. We have shown that it is equivalent to Π-injectivity, that it implies uniqueness of type and also that it is undecidable. Finally we have given two (partially) decidable criteria ensuring that subject reduction holds.

We believe that it is possible to give more general criteria. Several approaches could be considered for further research by, either refining our criterion by finding appropriate restrictions on rewrite rules producing product type, or designing new criteria for proving the confluence of the combination of a (non-left-linear) rewriting system with β-reduction. Another possibility would be to investigate the termination of the combination of a rewriting system with β-reduction without any assumption of confluence in order to prove confluence (hence subject reduction) using Newman's lemma.

10. REFERENCES

[1] T. Aoto, Y. Chiba, N. Hirokawa, and H. Zankl. Confluence competition (coco): http://coco.nue.riec.tohoku.ac.jp/index.php.

[2] F. Barbanera, M. Fernández, and H. Geuvers. Modularity of strong normalization in the algebraic-lambda-cube. *J. Funct. Program.*, 7(6):613–660, 1997.

[3] F. Blanqui. Definitions by rewriting in the Calculus of Constructions. *MSCS*, 2005.

[4] M. Boespflug, Q. Carbonneaux, O. Hermant, and R. Saillard. Dedukti: https://www.rocq.inria.fr/deducteam/dedukti.

[5] G. D. D. Cousineau. Embedding Pure Type Systems in λΠ-Calculus Modulo. In *TLCA*, 2007.

[6] Y. Matijasevich. Simple examples of undecidable associative calculi. *Doklady Mathematics*, 1967.

[7] M. H. A. Newman. On theories with a combinatorial definition of "equivalence". *Annals of Mathematics*, 43(2):pp. 223–243, 1942.

[8] R. Saillard. Towards explicit rewrite rules in the λΠ-calculus modulo. In *IWIL - 10th International Workshop on the Implementation of Logics*, 2013.

[9] Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

[10] L. S. van Benthem Jutting, J. McKinna, and R. Pollack. Checking algorithms for Pure Type Systems. In *Types for Proofs and Programs*. Springer Berlin Heidelberg, 1994.

[11] V. van Oostrom. *Confluence for Abstract and Higher-Order Rewriting*. PhD thesis, Vrije Universiteit, Amsterdam, 1994.