PROOF NORMALIZATION MODULO

GILLES DOWEK AND BENJAMIN WERNER

Abstract. We define a generic notion of cut that applies to many first-order theories. We prove a generic cut elimination theorem showing that the cut elimination property holds for all theories having a so-called pre-model. As a corollary, we retrieve cut elimination for several axiomatic theories, including Church's simple type theory.

Deduction modulo [11] is a formulation of first-order logic, that gives a formal account of the difference between deduction and computation in mathematical proofs. In deduction modulo a theory is formed with a set of axioms and a congruence, often defined by a set of rewrite rules. For instance, when defining arithmetic in deduction modulo, we can either take the usual axiom $\forall x\ x+0=x$ or orient this axiom as a rewrite rule $x+0\to x$, preserving provability. In the same way, when defining the theory of integral domains, we can chose to take the axiom $\forall x\ \forall y\ (x\times y=0\Leftrightarrow(x=0\vee y=0))$ or to orient this axiom as a rewrite rule $(x\times y=0)\to(x=0\vee y=0)$.

Axioms and rewrite rules play different roles in proofs and the structure of proofs is deeply affected when axioms are oriented as rewrite rules. In particular, the cut elimination property may hold for one formulation, but not for the other. Cut elimination is a much more powerful result (and hence more difficult to prove) when axioms are oriented as rewrite rules. In particular, for axiom free theories, cut elimination implies consistency as well as properties such as disjunction property and the witness property in the intuitionistic case. An important point is that deduction modulo permits to define a uniform notion of cut for all first-order theories that can be presented by rewrite rules only. This notion of cut subsumes that of Church's simple type theory (also called higher-order logic) since simple type theory can be defined as an axiom free first-order theory modulo. More generally, it subsumes the notion of cut introduced by Prawitz [27] and used for various theories, in particular for set theory [6, 18, 1, 7, 12] (see [10] for a discussion on this point).

This paper is an attempt to characterize the rewrite systems which define theories in deduction modulo for which cut elimination holds. The first part of the paper presents counter-examples to cut-elimination showing that termination and confluence are not sufficient conditions for the rewrite system to ensure cut elimination for the corresponding theory modulo. The second result is a generic cut elimination theorem, along the lines of Tait and Girard, showing that the cut elimination property holds, provided the congruence has a so-called pre-model. As a corollary, we obtain cut elimination for many theories, in particular for theories presented by a quantifier free confluent and terminating rewrite system,

for theories presented by a confluent and terminating positive rewrite system and for simple type theory. This way, we obtain a modular cut elimination proof for simple type theory where all the specificities of this theory are concentrated in the pre-model construction; the lemma that cut elimination holds in some theory if it has a pre-model remains completely general.

§1. Deduction modulo.

1.1. Natural deduction modulo and sequent calculus modulo. Deduction modulo gives a formal account of the difference between deduction and computation. This idea can be traced back to several sources, either in the field of equational unification and automated deduction [26, 14, 20, 28, 3], in the analysis of substitution in simple type theory [8, 4, 19, 27] or in modern type theories and λ-calculus [23, 22, 5, 21, 25, 31, 24].

This work is an attempt to study this distinction from a proof-theoretic point of view and independently of any particular application or formalism, *i.e.* in "plain" first-order logic.

More precisely, the notions of language, term, proposition are that of many-sorted first-order logic [13, 15]. We thus consider a set of *sorts*, an infinite set of *variables* of each sort, and a set of function symbols and of predicate symbols, that come with their *rank*. The formation rules for objects and propositions are the usual ones.

- Variables of sort s are terms of sort s.
- If f is a function symbol of rank $\langle s_1, \dots, s_n, s' \rangle$ and $t_1, \dots, t_n$ are respectively objects of sort $s_1, \dots, s_n$, then $f(t_1, \dots, t_n)$ is a term of sort s'.
- If P is a predicate symbol of rank $\langle s_1, \dots, s_n \rangle$ and $t_1, \dots, t_n$ are respectively objects of sort $s_1, \dots, s_n$, then $P(t_1, \dots, t_n)$ is an *atomic proposition*.

Propositions are built-up from atomic propositions with the usual connectors and quantifiers $\Rightarrow, \wedge, \vee, \bot, \forall$ and $\exists$. Remark that, implicitly, quantification in $\forall x\ P$ or $\exists x\ P$ is restricted over the sort of the variable x.

The characteristic of deduction modulo is that a *theory* is formed by a set of axioms Γ *and* a congruence $\equiv$ over propositions. Figure 1 gives the rules of natural deduction modulo and figure 2 those of sequent calculus modulo. As usual, intuitionistic natural deduction is obtained by dropping the excluded middle rule and intuitionistic sequent calculus by restricting to sequents with at most one conclusion; this requires a slight adaptation of the cut rule and the $\Rightarrow$-left rule.

$$\frac{\Gamma, A \vdash_{\equiv} \Delta \qquad \Gamma \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} \Delta}\ \text{cut if } A \equiv B$$

$$\frac{\Gamma \vdash_{\equiv} A \qquad \Gamma, B \vdash_{\equiv} \Delta}{\Gamma, C \vdash_{\equiv} \Delta}\ \Rightarrow\text{-left if } C \equiv (A \Rightarrow B)$$

Proposition 1.1 (Equivalence). *For every congruence $\equiv$ there is a set of axioms $\mathcal{T}$ such that $\Gamma \vdash_{\equiv} P$ if and only if $\mathcal{T}, \Gamma \vdash P$.*

Proof. Take for instance all the axioms of the form $\forall x_1\ \dots\ \forall x_n\ (P \Leftrightarrow Q)$ where $P \equiv Q$. ⊣

$$\frac{}{\Gamma \vdash_{\equiv} B}\ \text{axiom if } A \in \Gamma \text{ and } A \equiv B$$

$$\frac{\Gamma, A \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} C}\ \Rightarrow\text{-intro if } C \equiv (A \Rightarrow B)$$

$$\frac{\Gamma \vdash_{\equiv} C \qquad \Gamma \vdash_{\equiv} A}{\Gamma \vdash_{\equiv} B}\ \Rightarrow\text{-elim if } C \equiv (A \Rightarrow B)$$

$$\frac{\Gamma \vdash_{\equiv} A \qquad \Gamma \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} C}\ \wedge\text{-intro if } C \equiv (A \wedge B)$$

$$\frac{\Gamma \vdash_{\equiv} C}{\Gamma \vdash_{\equiv} A}\ \wedge\text{-elim if } C \equiv (A \wedge B)$$

$$\frac{\Gamma \vdash_{\equiv} C}{\Gamma \vdash_{\equiv} B}\ \wedge\text{-elim if } C \equiv (A \wedge B)$$

$$\frac{\Gamma \vdash_{\equiv} A}{\Gamma \vdash_{\equiv} C}\ \vee\text{-intro if } C \equiv (A \vee B)$$

$$\frac{\Gamma \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} C}\ \vee\text{-intro if } C \equiv (A \vee B)$$

$$\frac{\Gamma \vdash_{\equiv} D \qquad \Gamma, A \vdash_{\equiv} C \qquad \Gamma, B \vdash_{\equiv} C}{\Gamma \vdash_{\equiv} C}\ \vee\text{-elim if } D \equiv (A \vee B)$$

$$\frac{\Gamma \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} A}\ \bot\text{-elim if } B \equiv \bot$$

$$\frac{\Gamma \vdash_{\equiv} A}{\Gamma \vdash_{\equiv} B}\ \langle x, A\rangle\ \forall\text{-intro if } B \equiv (\forall x\ A) \text{ and } x \notin FV(\Gamma)$$

$$\frac{\Gamma \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} C}\ \langle x, A, t\rangle\ \forall\text{-elim if } B \equiv (\forall x\ A) \text{ and } C \equiv [t/x]A$$

$$\frac{\Gamma \vdash_{\equiv} C}{\Gamma \vdash_{\equiv} B}\ \langle x, A, t\rangle\ \exists\text{-intro if } B \equiv (\exists x\ A) \text{ and } C \equiv [t/x]A$$

$$\frac{\Gamma \vdash_{\equiv} C \qquad \Gamma, A \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} B}\ \langle x, A\rangle\ \exists\text{-elim if } C \equiv (\exists x\ A) \text{ and } x \notin FV(\Gamma B)$$

$$\frac{}{\Gamma \vdash_{\equiv} A}\ B \text{ excluded middle if } A \equiv (B \vee (B \Rightarrow \bot))$$

FIGURE 1. Natural deduction modulo

When the congruence $\equiv$ is decidable, proof-checking is decidable also, since we provide the information necessary for proof-checking in the quantifier rules. In many cases the congruence $\equiv$ is generated by a set of rewrite rules, when this rewrite system is terminating and confluent, the congruence $\equiv$ is decidable.

Like in usual natural deduction, a *cut* in a proof is an elimination rule, whose main premise is the introduction rule of the same connector or quantifier.

1.2. An example: Church's simple type theory. As mentioned above deduction modulo permits to express (intentional) simple type theory [4] without any axiom.

The *sorts* are *simple types* inductively defined by:

- ι and o are sorts,

$$\frac{}{A \vdash_{\equiv} B}\ \text{axiom if } A \equiv B$$

$$\frac{\Gamma, A \vdash_{\equiv} \Delta \qquad \Gamma \vdash_{\equiv} B, \Delta}{\Gamma \vdash_{\equiv} \Delta}\ \text{cut if } A \equiv B$$

$$\frac{\Gamma, B_1, B_2 \vdash_{\equiv} \Delta}{\Gamma, A \vdash_{\equiv} \Delta}\ \text{contr-left if } A \equiv B_1 \equiv B_2$$

$$\frac{\Gamma \vdash_{\equiv} B_1, B_2, \Delta}{\Gamma \vdash_{\equiv} A, \Delta}\ \text{contr-right if } A \equiv B_1 \equiv B_2$$

$$\frac{\Gamma \vdash_{\equiv} \Delta}{\Gamma, A \vdash_{\equiv} \Delta}\ \text{weak-left}$$

$$\frac{\Gamma \vdash_{\equiv} \Delta}{\Gamma \vdash_{\equiv} A, \Delta}\ \text{weak-right}$$

$$\frac{\Gamma \vdash_{\equiv} A, \Delta \qquad \Gamma, B \vdash_{\equiv} \Delta}{\Gamma, C \vdash_{\equiv} \Delta}\ \Rightarrow\text{-left if } C \equiv (A \Rightarrow B)$$

$$\frac{\Gamma, A \vdash_{\equiv} B, \Delta}{\Gamma \vdash_{\equiv} C, \Delta}\ \Rightarrow\text{-right if } C \equiv (A \Rightarrow B)$$

$$\frac{\Gamma, A, B \vdash_{\equiv} \Delta}{\Gamma, C \vdash_{\equiv} \Delta}\ \wedge\text{-left if } C \equiv (A \wedge B)$$

$$\frac{\Gamma \vdash_{\equiv} A, \Delta \qquad \Gamma \vdash_{\equiv} B, \Delta}{\Gamma \vdash_{\equiv} C, \Delta}\ \wedge\text{-right if } C \equiv (A \wedge B)$$

$$\frac{\Gamma, A \vdash_{\equiv} \Delta \qquad \Gamma, B \vdash_{\equiv} \Delta}{\Gamma, C \vdash_{\equiv} \Delta}\ \vee\text{-left if } C \equiv (A \vee B)$$

$$\frac{\Gamma \vdash_{\equiv} A, \Delta}{\Gamma \vdash_{\equiv} C, \Delta}\ \vee\text{-right1 if } C \equiv (A \vee B)$$

$$\frac{\Gamma \vdash_{\equiv} B, \Delta}{\Gamma \vdash_{\equiv} C, \Delta}\ \vee\text{-right2 if } C \equiv (A \vee B)$$

$$\frac{}{\Gamma, A \vdash_{\equiv} \Delta}\ \bot\text{-left if } A \equiv \bot$$

$$\frac{\Gamma, C \vdash_{\equiv} \Delta}{\Gamma, B \vdash_{\equiv} \Delta}\ \langle x, A, t\rangle\ \forall\text{-left if } B \equiv (\forall x\ A) \text{ and } C \equiv [t/x]A$$

$$\frac{\Gamma \vdash_{\equiv} A, \Delta}{\Gamma \vdash_{\equiv} B, \Delta}\ \langle x, A\rangle\ \forall\text{-right if } B \equiv (\forall x\ A) \text{ and } x \notin FV(\Gamma\Delta)$$

$$\frac{\Gamma, A \vdash_{\equiv} \Delta}{\Gamma, B \vdash_{\equiv} \Delta}\ \langle x, A\rangle\ \exists\text{-left if } B \equiv (\exists x\ A) \text{ and } x \notin FV(\Gamma\Delta)$$

$$\frac{\Gamma \vdash_{\equiv} C, \Delta}{\Gamma \vdash_{\equiv} B, \Delta}\ \langle x, A, t\rangle\ \exists\text{-right if } B \equiv (\exists x\ A) \text{ and } C \equiv [t/x]A$$

FIGURE 2. Sequent calculus modulo

- if T and U are sorts then $T \to U$ is a sort.

The language is composed of the individual symbols

- $S_{T,U,V}$ of sort $(T \to U \to V) \to (T \to U) \to T \to V$,
- $K_{T,U}$ of sort $T \to U \to T$,
- $\dot{\Rightarrow}$, $\dot{\wedge}$ and $\dot{\vee}$ of sort $o \to o \to o$, $\dot{\bot}$ of sort o,

- $\dot\forall_T$ and $\dot\exists_T$ of sort $(T \to o) \to o$,

the function symbols

- $\alpha_{T,U}$ of rank $\langle T \to U, T, U\rangle$,

and the predicate symbol

- ε of rank $\langle o\rangle$.

The combinators $S_{T,U,V}$ and $K_{T,U}$ are used to express functions. The objects $\dot\Rightarrow, \dot\wedge, \dot\vee, \dot\bot, \dot\forall_T$ and $\dot\exists_T$ allow to represent propositions as objects of sort o. Finally, the predicate ε allows to transform such an object t of type o into the actual corresponding proposition $\varepsilon(t)$. This typical reflection operation appears clearly in the rewrite rules.

$$\begin{aligned}
\alpha(\alpha(\alpha(S_{T,U,V}, x), y), z) &\to \alpha(\alpha(x, z), \alpha(y, z))\\
\alpha(\alpha(K_{T,U}, x), y) &\to x
\end{aligned}$$

$$\begin{aligned}
\varepsilon(\alpha(\alpha(\dot\Rightarrow, x), y)) &\to \varepsilon(x) \Rightarrow \varepsilon(y)\\
\varepsilon(\alpha(\alpha(\dot\wedge, x), y)) &\to \varepsilon(x) \wedge \varepsilon(y)\\
\varepsilon(\alpha(\alpha(\dot\vee, x), y)) &\to \varepsilon(x) \vee \varepsilon(y)\\
\varepsilon(\dot\bot) &\to \bot\\
\varepsilon(\alpha(\dot\forall, x)) &\to \forall y\ \varepsilon(\alpha(x, y))\\
\varepsilon(\alpha(\dot\exists, x)) &\to \exists y\ \varepsilon(\alpha(x, y))
\end{aligned}$$

1.3. Proof-terms. Cut-elimination can be viewed as a proof-transformation process. In order to study these transformations, we adopt a compact notation, representing proofs as λ-terms, for which the deduction rules act as typing rules; this is a typical use of the Curry-Howard isomorphism. In other words, we define a family of type systems parametrized by a congruence.

In this language, the proof-terms can contain both variables of the first-order language (written $x, y, \dots$) and proof variables (written $\alpha, \beta, \dots$). Terms of the first-order language are written $t, u, \dots$ while proof-terms are written $\pi, \rho, \dots$

Definition 1.1 (Proof-terms).

$$\begin{aligned}
\pi :=\ & \alpha\\
& \mid \lambda\alpha\ \pi \mid (\pi\ \pi')\\
& \mid \langle\pi, \pi'\rangle \mid \mathit{fst}(\pi) \mid \mathit{snd}(\pi)\\
& \mid i(\pi) \mid j(\pi) \mid (\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3)\\
& \mid (\delta_\bot\ \pi)\\
& \mid \lambda x\ \pi \mid (\pi\ t)\\
& \mid \langle t, \pi\rangle \mid (\delta_\exists\ \pi\ x\alpha\pi')
\end{aligned}$$

Notice that the variables α, β and x are bound in the constructions $\lambda\alpha\ \pi$, $\lambda x\ \pi$ $(\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3)$ and $(\delta_\exists\ \pi\ x\alpha\pi')$. Alphabetic equivalence is defined accordingly.

$$\frac{}{\Gamma \vdash_{\equiv} \alpha : B}\ \text{axiom if } \alpha : A \in \Gamma \text{ and } A \equiv B$$

$$\frac{\Gamma, \alpha : A \vdash_{\equiv} \pi : B}{\Gamma \vdash_{\equiv} \lambda\alpha\ \pi : C}\ \Rightarrow\text{-intro if } C \equiv (A \Rightarrow B)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : C \qquad \Gamma \vdash_{\equiv} \pi' : A}{\Gamma \vdash_{\equiv} (\pi\ \pi') : B}\ \Rightarrow\text{-elim if } C \equiv (A \Rightarrow B)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : A \qquad \Gamma \vdash_{\equiv} \pi' : B}{\Gamma \vdash_{\equiv} \langle \pi, \pi' \rangle : C}\ \wedge\text{-intro if } C \equiv (A \wedge B)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : C}{\Gamma \vdash_{\equiv} \mathit{fst}(\pi) : A}\ \wedge\text{-elim if } C \equiv (A \wedge B)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : C}{\Gamma \vdash_{\equiv} \mathit{snd}(\pi) : B}\ \wedge\text{-elim if } C \equiv (A \wedge B)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : A}{\Gamma \vdash_{\equiv} i(\pi) : C}\ \vee\text{-intro if } C \equiv (A \vee B)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : B}{\Gamma \vdash_{\equiv} j(\pi) : C}\ \vee\text{-intro if } C \equiv (A \vee B)$$

$$\frac{\Gamma \vdash_{\equiv} \pi_1 : D \qquad \Gamma, \alpha : A \vdash_{\equiv} \pi_2 : C \qquad \Gamma, \beta : B \vdash_{\equiv} \pi_3 : C}{\Gamma \vdash_{\equiv} (\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3) : C}\ \vee\text{-elim if } D \equiv (A \vee B)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : B}{\Gamma \vdash_{\equiv} (\delta_{\perp}\ \pi) : A}\ \perp\text{-elim if } B \equiv \perp$$

$$\frac{\Gamma \vdash_{\equiv} \pi : A}{\Gamma \vdash_{\equiv} \lambda x\ \pi : B}\ \langle x, A \rangle\ \forall\text{-intro if } B \equiv (\forall x\ A) \text{ and } x \notin FV(\Gamma)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : B}{\Gamma \vdash_{\equiv} (\pi\ t) : C}\ \langle x, A, t \rangle\ \forall\text{-elim if } B \equiv (\forall x\ A) \text{ or } C \equiv [t/x]A$$

$$\frac{\Gamma \vdash_{\equiv} \pi : C}{\Gamma \vdash_{\equiv} \langle t, \pi \rangle : B}\ \langle x, A, t \rangle\ \exists\text{-intro if } B \equiv (\exists x\ A) \text{ and } C \equiv [t/x]A$$

$$\frac{\Gamma \vdash_{\equiv} \pi : C \qquad \Gamma, \alpha : A \vdash_{\equiv} \pi' : B}{\Gamma \vdash_{\equiv} (\delta_{\exists}\ \pi\ x\alpha\pi') : B}\ \langle x, A \rangle\ \exists\text{-elim if } C \equiv (\exists x\ A) \text{ and } x \notin FV(\Gamma, B)$$

FIGURE 3. Typing rules

Each proof-term construction corresponds to a natural deduction rule: terms of the form α express proofs built with the axiom rule, terms of the form $\lambda\alpha\ \pi$ and $(\pi\ \pi')$ express proofs built respectively with the introduction and elimination rules of the implication, terms of the form $\langle \pi, \pi' \rangle$ and $\mathit{fst}(\pi)$, $\mathit{snd}(\pi)$ express proofs built with the introduction and elimination rules of the conjunction, terms of the form $i(\pi), j(\pi)$ and $(\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3)$ express proofs built with the introduction and elimination rules of the disjunction, terms of the form $(\delta_{\perp}\ \pi)$ express proofs built with the elimination rule of the contradiction, terms of the form $\lambda x\ \pi$ and $(\pi\ t)$ express proofs built with the introduction and elimination rules of the universal quantifier and terms of the form $\langle t, \pi \rangle$ and $(\delta_{\exists}\ \pi\ x\alpha\pi')$ express proofs built with the introduction and elimination rules of the existential quantifier.

$$
\begin{aligned}
(\lambda\alpha\ \pi_1\ \pi_2) &\to [\pi_2/\alpha]\pi_1\\
\mathit{fst}(\langle\pi_1,\pi_2\rangle) &\to \pi_1\\
\mathit{snd}(\langle\pi_1,\pi_2\rangle) &\to \pi_2\\
\delta(i(\pi_1),\alpha\pi_2,\beta\pi_3) &\to [\pi_1/\alpha]\pi_2\\
\delta(j(\pi_1),\alpha\pi_2,\beta\pi_3) &\to [\pi_1/\beta]\pi_3\\
(\lambda x\ \pi\ t) &\to [t/x]\pi\\
(\delta_\exists\ \langle t,\pi_1\rangle\ \alpha x\pi_2) &\to [t/x,\pi_1/\alpha]\pi_2
\end{aligned}
$$

FIGURE 4. Proof reduction rules

All this materializes in the typing rules of the calculus, given in figure 3. In this formulation, deduction modulo is a regular λ-calculus with dependent types, whose only originality is that types are identified modulo $\equiv$.

That provability is preserved in this presentation is quite straightforward.

PROPOSITION 1.2. *A sequent $A_1,\ldots,A_n \vdash_\equiv B$ is derivable in intuitionistic natural deduction modulo if and only if there exists a term π such that the judgment $\alpha_1 : A_1,\ldots,\alpha_n : A_n \vdash_\equiv \pi : B$ is derivable in the system of figure 3.*

In the definition of such a calculus some usual practical choices have to be made. Here we have chosen a λ-calculus *à la Curry*, *i.e.* where abstracted proof variables are not explicitly typed and where the same proof-term $\lambda\alpha\ \alpha$ is a proof of the proposition $A \Rightarrow A$ and of the proposition $B \Rightarrow B$. We have also chosen to define the proof-terms first and then the well-typed proof-terms. So a proof-term does not need to be well-typed (*i.e.* to correspond to a meaningful proof).

1.4. Proof reduction rules. As usual, the process of cut elimination is modeled by (generalized) β-reduction. We consider the contextual closure of the reduction rules given figure 4. These rules correspond to proof reduction in natural deduction. In order to obtain cut elimination in sequent calculus, we will have to consider their extension to the so-called reductions of commutative cuts in subsection 3.6.

We write $\pi \to^1 \pi'$ if π reduces in one step to π', $\pi \to^+ \pi'$ if π reduces in at least one step to π', and $\pi \to \pi'$ if π reduces in an arbitrary number of steps to π'.

A proof is said to be *normal* if it contains no redex. It is said to be *normalizing* if it has a normal form and *strongly normalizing* if all reduction sequences issued from this proofs are finite. We write $\mathcal{SN}$ for the set of strongly normalizing proofs.

PROPOSITION 1.3 (Subject reduction). *If $\Gamma \vdash_\equiv \pi : P$ and $\pi \to \pi'$ then $\Gamma \vdash_\equiv \pi' : P$.*

THEOREM 1.1. *Provided $\equiv$ is defined by a confluent rewrite system, which rewrites terms to terms and atomic propositions to propositions, there is no normal proof of the sequent $\vdash_\equiv \bot$.*

PROOF. The properties of the rewrite system ensure that $\bot$ and any other non-atomic proposition (like $A \wedge B$) have no common reduct. Thus, by confluence, they are not congruent.

By induction over proof-term structure, one easily checks that a normal closed proof can only end with an introduction rule; *i.e.* the proof-term has the form $\lambda\alpha\ \pi$, $\langle \pi, \pi' \rangle$, $i(\pi)$, $j(\pi)$, $\lambda x\ \pi$, $\langle t, \pi \rangle$. By the remark above, it is obvious that none of these terms can be a proof of $\bot$. $\dashv$

1.5. Deduction modulo and conversion rules. An alternative presentation of deduction modulo, would be to keep the usual deduction rule and add a conversion rule

$$\frac{\Gamma \vdash_{\equiv} A}{\Gamma \vdash_{\equiv} B} \text{ if } A \equiv B$$

Although both presentations are obviously equivalent from the provability point of view, this latter presentation obscures the notion of cut, as a cut needs now to be defined as a sequence formed by

- an introduction,
- an arbitrary number of conversions,
- and an elimination

(see for instance [9]).

When considering proof-terms, it is usual in modern type theory ([22, 5, 2, 25, 31]) to take the conversion rule

$$\frac{\Gamma \vdash_{\equiv} \pi : A}{\Gamma \vdash_{\equiv} \pi : B} \text{ if } A \equiv B$$

where the conversion steps are not recorded in the proof-terms. In this case the reduction rules are those of figure 4, but this bears a discrepancy between the structure of the proof and that of the term.

An alternative would be to introduce a *cast* construction and to take the rule

$$\frac{\Gamma \vdash_{\equiv} \pi : A}{\Gamma \vdash_{\equiv} (cast_B\ \pi) : B} \text{ if } A \equiv B$$

In this case, like above, a redex must be defined as a term formed by

- an introduction,
- an arbitrary number of casts,
- and an elimination.

§2. Counter-examples to cut elimination. For some congruences, deduction modulo enjoys cut elimination, while it does not for others. We provide, in this section, examples of congruences modulo which proof normalization fails.

In particular it may be noticed that normalization may fail even when the congruence is defined by a confluent and terminating rewrite system.

2.1. Russell's paradox. Consider the following rewrite rule

$$R \ \rightarrow \ (R \Rightarrow S)$$

Modulo this rewrite rule, the proof $\lambda\alpha\ (\alpha\ \alpha)\ \lambda\alpha\ (\alpha\ \alpha)$ has type S. The only way to reduce this proof is to reduce it to itself and hence it is not normalizable.

An instance of this rewrite rule is skolemized naive set theory. In naive set theory we have the following axiom scheme

$$\forall x_1 \ \ldots \ \forall x_n \ \exists y \ \forall z \ (z \in y \Leftrightarrow P)$$

for any propositional expression P. Skolemizing this scheme, we introduce for each proposition P a symbol $f_{x_1,\ldots,x_n,z,P}$ and an axiom

$$\forall x_1 \ \ldots \ \forall x_n \ \forall z \ (z \in f_{x_1,\ldots,x_n,z,P}(x_1,\ldots,x_n) \Leftrightarrow P)$$

This axiom can be turned into the rewrite rule

$$z \in f_{x_1,\ldots,x_n,z,P}(x_1,\ldots,x_n) \ \rightarrow \ P$$

In particular we have a rewrite rule

$$z \in f_{z,(z\in z)\Rightarrow\bot} \ \rightarrow \ (z \in z) \Rightarrow \bot$$

and hence writing R for the proposition $f_{z,(z\in z)\Rightarrow\bot} \in f_{z,(z\in z)\Rightarrow\bot}$ and S for the proposition $\bot$ we have

$$R \ \rightarrow \ (R \Rightarrow S)$$

2.2. Crabbé's counter-example. Even if Zermelo's set theory is considered consistent, it is well-known that cut elimination is problematic and does generally not hold. The proof of non normalization is called Crabbé's counter-example [6, 18, 12].

Consider the following (non terminating) rewrite rule

$$C \ \rightarrow \ E \wedge (C \Rightarrow D)$$

Modulo this rewrite rule, the proof

$$\lambda\alpha \ (\mathit{snd}(\alpha) \ \alpha) \ \langle\beta, \lambda\alpha \ (\mathit{snd}(\alpha) \ \alpha)\rangle$$

is a proof of D in the context E. The only way to reduce this proof is to reduce it to

$$(\mathit{snd}(\langle\beta, \lambda\alpha \ (\mathit{snd}(\alpha) \ \alpha)\rangle) \ \langle\beta, \lambda\alpha \ (\mathit{snd}(\alpha) \ \alpha)\rangle)$$

and then to itself

$$(\lambda\alpha \ (\mathit{snd}(\alpha) \ \alpha) \ \langle\beta, \lambda\alpha \ (\mathit{snd}(\alpha) \ \alpha)\rangle)$$

Hence it is not normalizable.

An instance of such a rule is skolemized set theory. In set theory, we have an axiom scheme

$$\forall x_1 \ \ldots \ \forall x_n \ \forall w \ \exists y \ \forall z \ (z \in y \Leftrightarrow (z \in w \wedge P))$$

skolemizing this scheme, we introduce for each proposition P a symbol $f_{x_1,\ldots,x_n,z,P}$ and an axiom

$$\forall x_1 \ \ldots \ \forall x_n \ \forall w \ \forall z \ (z \in f_{x_1,\ldots,x_n,z,P}(x_1,\ldots,x_n,w) \Leftrightarrow (z \in w \wedge P))$$

This axiom can be turned into the rewrite rule

$$z \in f_{x_1,\ldots,x_n,z,P}(x_1,\ldots,x_n,w) \ \rightarrow \ z \in w \wedge P$$

In particular we have a rewrite rule

$$z \in f_{z,(z\in z)\Rightarrow\bot}(w) \ \rightarrow \ z \in w \wedge (z \in z \Rightarrow \bot)$$

and hence writing C for the proposition $f_{z,(z\in z)\Rightarrow\bot}(w) \in f_{z,(z\in z)\Rightarrow\bot}(w)$, D for the proposition $\bot$ and E for the proposition $f_{z,(z\in z)\Rightarrow\bot}(w) \in w$ we have

$$C \ \rightarrow \ E \wedge (C \Rightarrow D)$$

2.3. A terminating presentation of Russell's paradox. This is the most interesting counter-example, since is takes place in a theory defined by a confluent and terminating rewrite system. It is a refined version of the example of section 2.1 where termination is obtained by adding a condition in the right-hand part which freezes the redex.

Let $\neg A$ stand for $A \Rightarrow \bot$. Instead of taking the (non terminating) rule $R \in R \ \rightarrow \ \neg R \in R$, we take the (terminating) rule

$$R \in R \ \rightarrow \ \forall y \ (y \simeq R \Rightarrow \neg y \in R)$$

where $\simeq$ must be chosen such that $R \simeq R$ holds and we can prove $R \in R$ from $R \in y$ and $y \simeq R$.

One possibility is to chose $y \simeq z$ to stand for $\forall x \ (y \in x \Rightarrow z \in x)$ (which is a possible set-theoretical definition of Leibniz equality). This leads to the proofs:

$$\begin{array}{rcll} \pi & = & \lambda\alpha^{R\in R} \ (\alpha \ R \ (\lambda x \ \lambda\beta^{R\in x}\beta) \ \alpha) & : \ \neg R \in R \\ \pi' & = & \lambda y \ \lambda\alpha^{\forall x.y\in x\Rightarrow R\in x} \ \lambda\beta^{y\in R} \ (\pi \ (\alpha \ R \ \beta)) & : \ R \in R \end{array}$$

We have $(\pi \ \pi') : \bot$ hence the theory is inconsistent. Moreover, this proof does not normalize. Indeed the proposition $\bot$ has no cut free proof. Thus, the theory does not verify the cut elimination property.

§3. Proof Normalization in Natural Deduction. Let us now identify some congruences for which normalization holds. We want to prove that proofs normalize modulo congruences defined by a confluent and terminating quantifier free rewrite system, (as in the example $x \times y = 0 \rightarrow x = 0 \vee y = 0$), modulo congruences defined by a confluent and terminating positive rewrite system, *i.e.* one containing no negative occurrences of atomic propositions and modulo the congruence defining simple type theory above. All these results are consequences of a unique theorem stating that deduction modulo a congruence has the normalization property if this congruence has a *pre-model*.

3.1. Ultra-reduction. In order to allow the lift of the normalization theorem from natural deduction to natural deduction with commutative cuts and to the sequent calculus, we need to generalize slightly the result and prove strong normalization for *ultra-reductions*.

Ultra-reductions are inspired by Girard [16], and are obtained by adding to the rules of figure 4 three more rules yielding the rules of figure 5. We keep the notation $\rightarrow$ for reduction and we write $\triangleright$ for ultra-reduction. Obviously strong normalization of ultra-reduction implies that of reduction. On the other hand, confluence is obviously lost.

A technical point is that some variables may get freed in the reduct. A rigorous presentation would involve introducing a fresh variable for each occurrence of an ultra-reduction in order to avoid capture of these variables. Moreover subject reduction is not preserved for these reductions. A fully safe presentation of

$$(\lambda\alpha\ \pi_1\ \pi_2) \rhd [\pi_2/\alpha]\pi_1$$
$$fst(\langle\pi_1,\pi_2\rangle) \rhd \pi_1$$
$$snd(\langle\pi_1,\pi_2\rangle) \rhd \pi_2$$
$$\delta(i(\pi_1),\alpha\pi_2,\beta\pi_3) \rhd [\pi_1/\alpha]\pi_2$$
$$\delta(j(\pi_1),\alpha\pi_2,\beta\pi_3) \rhd [\pi_1/\beta]\pi_3$$
$$(\lambda x\ \pi\ t) \rhd [t/x]\pi$$
$$(\delta_\exists\ \langle t,\pi_1\rangle\ \alpha x\pi_2) \rhd [t/x,\pi_1/\alpha]\pi_2$$

$$(\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3) \rhd \pi_2$$
$$(\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3) \rhd \pi_3$$
$$(\delta_\exists\ \pi_1\ x\alpha\pi_2) \rhd \pi_2$$

FIGURE 5. Ultra-reduction rules

ultra-reductions would be to replace the freed variables by a dummy proof:

$$(\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3) \rhd \pi_2[(\delta_\bot\gamma_0)/\alpha]$$

with γ_0 a proof variable of type $\bot$ in the context.

Since subject reduction is not really an issue here, we stick to the presentation of figure 5 and disregard the variable capture problems.

3.2. Reducibility. The basic tool used hereafter is reducibility; the main concepts are due to Tait [29] and Girard [16, 17]. In particular, since we want to treat the case of Higher-Order Logic, we need some form of reducibility candidates. We here chose a definition similar to Girard's [17], but other ones like Tait's saturated sets would also apply [30].

DEFINITION 3.1 (Neutral proof-term). *A proof-term is said to be* neutral *if it is an axiom or an elimination (*i.e. α, $(\pi\ t)$, $(\pi\ \pi')$, $fst(\pi)$, $snd(\pi)$, $(\delta\ \pi\ \alpha\pi_1\ \beta\pi_2)$, $(\delta_\bot\ \pi)$, $(\delta_\exists\ \pi_1\ x\alpha\pi_2)$*).*

DEFINITION 3.2 (Reducibility candidate). *A set R of proof-terms is a* reducibility candidate *if*

- *if $\pi \in R$, then π is strongly normalizable,*
- *if $\pi \in R$ and $\pi \rhd \pi'$ then $\pi' \in R$,*
- *if π is neutral and if for every π' such that $\pi \rhd^1 \pi'$, $\pi' \in R$ then $\pi \in R$.*

Let $\mathcal{C}$ be the set of all reducibility candidates.

Mostly, we follow the main scheme of reducibility proofs. That is we try to construct for every proposition A a set of proof-terms $\mathcal{R}_A$ such that

- all elements of $\mathcal{R}_A$ are strongly normalizing,
- and if $\Gamma \vdash_\equiv \pi : A$ holds, then $\pi \in \mathcal{R}_A$.

The first condition is ensured by verifying that all $\mathcal{R}_A$ are reducibility candidates. The second one is proved by induction over the derivation of $\Gamma \vdash_\equiv \pi : A$ using closure conditions due to the definition of $\mathcal{R}_A$. Typically:

- If $\pi \in \mathcal{R}_{A\Rightarrow B}$ and π reduces to a proof-term of the form $\lambda\alpha\ \pi_1$, then for every proof-term π' of $\mathcal{R}_A$, $[\pi'/\alpha]\pi_1$ is an element of $\mathcal{R}_B$.
- If $\pi \in \mathcal{R}_{A\wedge B}$ and π reduces to a proof-term of the form $\langle\pi_1, \pi_2\rangle$, then π_1 is an element of $\mathcal{R}_A$ and π_2 is an element of $\mathcal{R}_B$.
- If $\pi \in \mathcal{R}_{A\vee B}$ and π reduces to a proof-term of the form $i(\pi_1)$ (resp. $j(\pi_2)$) then π_1 is an element of $\mathcal{R}_A$ (resp. π_2 is an element of $\mathcal{R}_B$).
- If $\pi \in \mathcal{R}_{\forall x\ A}$ and π reduces to a proof-term of the form $\lambda x\ \pi_1$ then for every term t of the same sort as x, $[t/x]\pi_1$ is an element of $\mathcal{R}_{[t/x]A}$.
- If $\pi \in \mathcal{R}_{\exists x\ A}$ and π reduces to a proof-term of the form $\langle t, \pi_1\rangle$ then $[t/x]\pi_1$ is an element of $\mathcal{R}_{[t/x]A}$.

If we read the conditions above as a partial definition of the family of sets $\mathcal{R}_A$, we understand that the crucial step is the choice of the sets $\mathcal{R}_A$ when A is atomic. In first-order logic, we usually take $\mathcal{SN}$ for the set $\mathcal{R}_A$ when A is atomic, but this is not possible here. Indeed, an atomic proposition A can be reduced to a non atomic one, say $B \Rightarrow C$. Thus a proof-term π_1 of A may be applied to a proof-term π_2 of B to form a proof-term $(\pi_1\ \pi_2)$ of C. The strong normalization of $(\pi_1\ \pi_2)$ cannot be deduced from that of π_1 and that of π_2 because the proof-term $(\pi_1\ \pi_2)$ itself may be a redex. Hence, as A rewrites to $B \Rightarrow C$ we have to take the same conditions on $\mathcal{R}_A$ than on $\mathcal{R}_{B\Rightarrow C}$. A solution is to take $\mathcal{R}_A = \mathcal{R}_{B\Rightarrow C}$. More generally we require that $\mathcal{R}_A = \mathcal{R}_B$ when $A \equiv B$.

To define the family $\mathcal{R}_A$, for A atomic, it is enough to define for every n-ary predicate symbol P the sets $\mathcal{R}_{P(t_1,\dots,t_n)}$, or equivalently to give a function $\hat{P}$ that maps n-tuples of terms to some well-chosen reducibility candidate.

It is well-known that a reducibility proof essentially boils down to the construction of a particular syntactical model. This comparison is particularly striking here since, in first-order logic, to define a model, we also need to provide, for every predicate symbol P a function $\hat{P}$ that maps every n-tuple of terms to a truth value.

We can pursue this comparison. If two terms t_1 and t'_1 are congruent then the sets $\hat{P}(t_1, \dots, t_n)$ and $\hat{P}(t'_1, \dots, t_n)$ must be identical. The function $\hat{P}$ is then better defined as a function from an abstract object (for instance, the class of t_1 and t'_1) that t_1 and t'_1 denote.

Then the condition that two congruent propositions must have the same denotation can be expressed as the fact that the congruence is valid in the model.

3.3. Pre-model. Formalizing the discussion above, we end-up with the following notion.

Definition 3.3 (Pre-model). *Let $\mathcal{L}$ be a (many sorted) first-order language. A* pre-model *for $\mathcal{L}$ is given by:*

- *for every sort T, a set $\mathcal{M}_T$,*
- *for every function symbol f of rank $\langle T_1, \dots, T_n, U\rangle$, a mapping $\hat{f}$ from the set $\mathcal{M}_{T_1} \times \cdots \times \mathcal{M}_{T_n}$ to the set $\mathcal{M}_U$,*
- *for every predicate symbol P of rank $\langle T_1, \dots, T_n\rangle$, a mapping $\hat{P}$ from the set $\mathcal{M}_{T_1} \times \cdots \times \mathcal{M}_{T_n}$ to the set $\mathcal{C}$ of reducibility candidates.*

DEFINITION 3.4. *Let t be a term and φ an assignment mapping all the free variables of t of sort T to elements of $\mathcal{M}_T$. We define the object $|t|_\varphi$ by induction over the structure of t.*

- $|x|_\varphi = \varphi(x)$,
- $|f(t_1, \ldots, t_n)|_\varphi = \hat{f}(|t_1|_\varphi, \ldots, |t_n|_\varphi)$.

DEFINITION 3.5. *Let A be a proposition and φ an assignment mapping all the free variables of A of sort T to elements of $\mathcal{M}_T$. We define the set $|A|_\varphi$ of proof-terms by induction over the structure of A.*

- *A proof-term is an element of $|P(t_1, \ldots, t_n)|_\varphi$ if it is in $\hat{P}(|t_1|_\varphi, \ldots, |t_n|_\varphi)$.*
- *A proof-term is an element of $|A \Rightarrow B|_\varphi$ if it is strongly normalizable and when it reduces to a proof-term of the form $\lambda\alpha\ \pi_1$ then for every π' in $|A|_\varphi$, $[\pi'/\alpha]\pi_1$ is an element of $|B|_\varphi$.*
- *A proof-term is an element of $|A \wedge B|_\varphi$ if it is strongly normalizable and when it reduces to a proof-term of the form $\langle \pi_1, \pi_2 \rangle$ then π_1 and π_2 are elements of $|A|_\varphi$ and $|B|_\varphi$.*
- *A proof-term is an element of $|A \vee B|_\varphi$ if it is strongly normalizable and when it reduces to a proof-term of the form $i(\pi_1)$ (resp. $j(\pi_2)$) then π_1 (resp. π_2) is an element of $|A|_\varphi$ (resp. $|B|_\varphi$).*
- *A proof-term is an element of $|\bot|_\varphi$ if it is strongly normalizable.*
- *A proof-term is an element of $|\forall x\ A|_\varphi$ if it is strongly normalizable and when it reduces to a proof-term of the form $\lambda x\ \pi_1$ then for every term t of sort T (where T is the sort of x) and every element E of $\mathcal{M}_T$, $[t/x]\pi_1$ is an element of $|A|_{\varphi + \langle x, E \rangle}$, where $\varphi + \langle x, u \rangle$ is the function that coincides with φ everywhere except in x where it takes value u.*
- *A proof-term is an element of $|\exists x\ A|_\varphi$ if it is strongly normalizable and when it reduces to a proof-term of the form $\langle t, \pi_1 \rangle$ there exists an element E of $\mathcal{M}_T$ (where T is the sort of x) such that π_1 is an element of $|A|_{\varphi + \langle x, E \rangle}$.*

DEFINITION 3.6. *A pre-model is a pre-model of $\equiv$ if, whenever $A \equiv B$, then for every assignment φ, $|A|_\varphi = |B|_\varphi$.*

PROPOSITION 3.1. *For every proposition A and assignment φ, $|A|_\varphi$ is a reducibility candidate*

PROOF. By induction over the structure of A.

If A is an atomic proposition, $|A|_\varphi$ is a reducibility candidate by definition.

If A is a composed proposition, then, by definition, $|A|_\varphi$ contains only normalizable proof-terms.

It is routine to prove closure by reduction.

Now, we assume that π is a neutral proof-term and for every π' such that $\pi \rhd^1 \pi'$, $\pi' \in |A|_\varphi$; we want to prove that π is in $|A|_\varphi$. Following the definition of $|A|_\varphi$, we first prove that π is strongly normalizable and then that if it reduces to an introduction, the subproofs belong to the appropriate sets.

We first prove that π is strongly normalizable. Let $\pi = \pi_1, \pi_2, \ldots$ be a reduction sequence issued from π. If this sequence is empty it is finite. Otherwise we have $\pi \rhd^1 \pi_2$ and hence π_2 is an element of $|A|_\varphi$ thus it is strongly normalizable and the reduction sequence is finite.

Then we prove that if π reduces to a introduction then the subproofs belong to the appropriate sets. Let $\pi = \pi_1, \pi_2, \ldots, \pi_n$ be a reduction sequence issued from π and such that π_n is an introduction. This sequence cannot be empty because π is neutral and hence not an introduction. Thus $\pi \rhd^1 \pi_2 \rhd \pi_n$. We have $\pi_2 \in |A|_\varphi$ and thus if π_n is an introduction the subproofs belong to the appropriate sets. ⊣

PROPOSITION 3.2 (Substitution). *Given any proposition A, term t and variable x we have*

$$|[t/x]A|_\varphi = |A|_{\varphi + \langle x, |t|_\varphi \rangle}$$

PROOF. By induction on the structure of A. ⊣

3.4. The normalization theorem. In this section we prove that if a system bears a pre-model then proof-terms modulo this system normalize.

THEOREM 3.1. *Let $\equiv$ be a congruence, $\mathcal{M}$ be a pre-model of $\equiv$, $\Gamma \vdash_\equiv \pi : A$ be derivable judgment. Consider:*

- *θ a substitution mapping the variables of any sort to terms of the same sort,*
- *φ an assignment mapping variables of any sort T to elements of $\mathcal{M}_T$,*
- *σ a substitution mapping any proof variable bound to proposition B in Γ to an element of $|B|_\varphi$. Then $\sigma\theta\pi$ is an element of $|A|_\varphi$.*

PROOF. By induction over the structure of π. We detail the various cases.

3.4.1. *Axiom.* If π is a variable α, we have $(\alpha : B) \in \Gamma$ with $A \equiv B$. By definition $\sigma\theta\alpha = \sigma\alpha$ which is an element of $|B|_\varphi$. Since $\mathcal{M}$ is a pre-model, $|A|_\varphi = |B|_\varphi$ and $\sigma\theta\alpha \in |A|_\varphi$.

3.4.2. *$\Rightarrow$-intro.* The proof-term π has the form $\lambda\alpha\ \rho$ where α is a proof variable of some proposition B and ρ a proof of some proposition C. We have $\sigma\theta\pi = \lambda\alpha\ \sigma\theta\rho$, consider a reduction sequence issued from this proof-term. This sequence can only reduce the proof-term $\sigma\theta\rho$. By induction hypothesis, the proof-term $\sigma\theta\rho$ is an element of $|C|_\varphi$, thus the reduction sequence is finite.

Furthermore, every reduct of $\sigma\theta\pi$ is of the form $\lambda\alpha\ \rho'$ where ρ' is a reduct of $\sigma\theta\rho$. Let then τ be any proof of $|B|_\varphi$, the proof-term $[\tau/\alpha]\rho'$ can be obtained by reduction from $([\tau/\alpha] \circ \sigma)\theta\rho$. By induction hypothesis, the proof-term $([\tau/\alpha] \circ \sigma)\theta\rho$ is an element of $|C|_\varphi$. Hence, as $|C|_\varphi$ is a reducibility candidate, the proof-term $[\tau/\alpha]\rho'$ is an element of $|C|_\varphi$.

Hence, the proof-term $\sigma\theta\lambda\alpha\ \rho$ is an element of $|A|_\varphi$.

3.4.3. *$\wedge$-intro.* The proof-term π has the form $\langle \rho_1, \rho_2 \rangle$ where ρ_1 is a proof of some proposition B and ρ_2 a proof of some proposition C. We have $\sigma\theta\pi = \langle \sigma\theta\rho_1, \sigma\theta\rho_2 \rangle$. Consider a reduction sequence issued from this proof-term. This sequence can only reduce the proof-terms $\sigma\theta\rho_1$ and $\sigma\theta\rho_2$. By induction hypothesis these proof-terms are in $|B|_\varphi$ and $|C|_\varphi$. Thus the reduction sequence is finite.

Furthermore, any reduct of $\sigma\theta\pi$ is of the form $\langle \rho'_1, \rho'_2 \rangle$ where ρ'_1 is a reduct of $\sigma\theta\rho_1$ and ρ'_2 one of ρ_2. These proof-terms are in $|B|_\varphi$ and $|C|_\varphi$ because these sets are candidates.

Hence, the proof-term $\sigma\theta\langle \rho_1, \rho_2 \rangle$ is in $|A|_\varphi$.

3.4.4. $\vee$-*intro.* The proof-term π has the form $i(\rho)$ (resp. $j(\rho)$) and ρ is a proof of some proposition B. We have $\sigma\theta\pi = i(\sigma\theta\rho)$ (resp. $j(\sigma\theta\rho)$). Consider a reduction sequence issued from this proof-term. This sequence can only reduce the proof-terms $\sigma\theta\rho$. By induction hypothesis this proof-term is an element of $|B|_\varphi$. Thus the reduction sequence is finite.

Furthermore all reducts of $\sigma\theta\pi$ are of the form $i(\rho')$ (resp. $j(\rho')$) where ρ' is a reduct of $\sigma\theta\rho$. Then ρ' is an element of $|B|_\varphi$ since this set is a candidate.

Hence, the proof-term $\sigma\theta i(\rho)$ (respectively $\sigma\theta j(\rho)$) is an element of $|A|_\varphi$.

3.4.5. $\forall$-*intro.* The proof-term π has the form $\lambda x\ \rho$ where ρ is a proof of some proposition B. We have $\sigma\theta\pi = \lambda x\ \sigma\theta\rho$.

Consider a reduction sequence issued from the proof-term $\sigma\theta\pi = \lambda x\ \sigma\theta\rho$. This sequence can only reduce the proof-term $\sigma\theta\rho$. Let E be an element of $\mathcal{M}_T$ (where T is the sort of x). By induction hypothesis, the proof-term $\sigma\theta\rho$ is an element of $|B|_{\varphi+\langle x,E\rangle}$, thus the reduction sequence is finite.

All reducts of $\sigma\theta\pi$ are of the form $\lambda x\ \rho'$ where ρ' is a reduct of $\sigma\theta\rho$. The proof-term $[t/x]\rho'$ is obtained by reducing the proof-term $[t/x]\sigma\theta\rho$. By induction hypothesis again, the proof-term $[t/x]\sigma\theta\rho$ is an element of $|B|_{\varphi+\langle x,E\rangle}$.

Hence $\sigma\theta\lambda x\ \rho$ is an element of $|A|_\varphi$.

3.4.6. $\exists$-*intro.* The proof-term π has the form $\langle t, \rho\rangle$, $A \equiv \exists x\ B$ and ρ is a proof of $[t/x]B$. We have $\sigma\theta\pi = \langle \theta t, \sigma\theta\rho\rangle$. Consider a reduction sequence issued from this proof-term. This sequence can only reduce the proof-term $\sigma\theta\rho$. By induction hypothesis this proof-term is in $|[t/x]B|_\varphi$. Thus the reduction sequence is finite.

Furthermore, let $E = |t|_\varphi$. Any reduct of $\sigma\theta\pi$ is of the form $\langle \theta t, \rho'\rangle$ where ρ' is a reduct of $\sigma\theta\rho$. This proof-term is an element of $|[t/x]B|_\varphi$, *i.e.* of $|B|_{\varphi+\langle x,E\rangle}$, because $|B|_{\varphi+\langle x,E\rangle}$ is a candidate.

Hence, the proof-term $\sigma\theta\langle t, \rho\rangle$ is an element of $|A|_\varphi$.

3.4.7. $\Rightarrow$-*elim.* The proof-term π has the form $(\rho_1\ \rho_2)$ and ρ_1 is a proof of some proposition $B \Rightarrow A$ and ρ_2 a proof of the proposition B. We have $\sigma\theta\pi = (\sigma\theta\rho_1\ \sigma\theta\rho_2)$. By induction hypothesis $\sigma\theta\rho_1$ and $\sigma\theta\rho_2$ are in the sets $|B \Rightarrow A|_\varphi$ and $|B|_\varphi$. Hence these proof-terms are strongly normalizable. Let n be the maximum length of a reduction sequence issued from $\sigma\theta\rho_1$ and n' the maximum length of a reduction sequence issued from $\sigma\theta\rho_2$. We prove by induction on $n + n'$ that $(\sigma\theta\rho_1\ \sigma\theta\rho_2)$ is in the set $|A|_\varphi$. Since this proof-term is neutral we only need to prove that every of its one step reducts is in $|A|_\varphi$. If the reduction takes place in $\sigma\theta\rho_1$ or in $\sigma\theta\rho_2$ then we apply the induction hypothesis. Otherwise $\sigma\theta\rho_1$ has the form $\lambda\alpha\ \rho'$ and the reduct is $[\sigma\theta\rho_2/\alpha]\rho'$. By the definition of $|B \Rightarrow A|_\varphi$ this proof-term is in $|A|_\varphi$.

Hence, the proof-term $\sigma\theta(\rho_1\ \rho_2)$ is an element of $|A|_\varphi$.

3.4.8. $\wedge$-*elim.* We only detail the case of left elimination. The proof-term π has the form $\mathit{fst}(\rho)$ where ρ is a proof of some proposition $A \wedge B$. We have $\sigma\theta\pi = \mathit{fst}(\sigma\theta\rho)$. By induction hypothesis the proof-term $\sigma\theta\rho$ is in $|A \wedge B|_\varphi$. Hence, it is strongly normalizable. Let n be the maximum length of a reduction sequence issued from this proof-term. We prove by induction on n that $\mathit{fst}(\sigma\theta\rho)$ is in the set $|A|_\varphi$. Since this proof-term is neutral we only need to prove that every of its one step reducts is in $|B|_\varphi$. If the reduction takes place in $\sigma\theta\rho$ then

we apply the induction hypothesis. Otherwise $\sigma\theta\rho$ has the form $\langle\rho'_1, \rho'_2\rangle$ and the reduct is ρ'_1. By the definition of $|A \wedge B|_\varphi$ this proof-term is in $|A|_\varphi$.

Hence, the proof-term $\sigma\theta fst(\rho)$ is an element of $|A|_\varphi$.

3.4.9. $\vee$-*elim.* The proof-term π has the form $(\delta\ \rho_1\ \alpha\rho_2\ \beta\rho_3)$ where ρ_1 is a proof-term of some proposition $B \vee C$ and ρ_2 and ρ_3 are proof-terms of A. We have $\sigma\theta\pi = (\delta\ \sigma\theta\rho_1\ \alpha\sigma\theta\rho_2\ \beta\sigma\theta\rho_3)$. By induction hypothesis, the proof-term $\sigma\theta\rho_1$ is in the set $|B \vee C|_\varphi$, and the proof-terms $\sigma\theta\rho_2$ and $\sigma\theta\rho_3$ are in the set $|A|_\varphi$. Hence, these proof-terms are strongly normalizable. Let n, n' and n'' be the maximum length of reduction sequences issued from these proof-terms. We prove by induction on $n + n' + n''$ that $(\delta\ \sigma\theta\rho_1\ \sigma\theta\alpha\rho_2\ \beta\sigma\theta\rho_3)$ is in $|A|_\varphi$. Since this proof-term is neutral we only need to prove that every of its one step reducts is in $|A|_\varphi$. If the reduction takes place in $\sigma\theta\rho_1$, $\sigma\theta\rho_2$ or $\sigma\theta\rho_3$ then we apply the induction hypothesis. Otherwise, if $\sigma\theta\rho_1$ has the form $i(\rho')$ (resp. $j(\rho')$) and the reduct is $[\rho'/\alpha]\sigma\theta\rho_2$ (resp. $[\rho'/\beta]\sigma\theta\rho_3$). By the definition of $|B \vee C|_\varphi$ the proof-term ρ' is in $|B|_\varphi$ (resp. $|C|_\varphi$). Hence by induction hypothesis $([\rho'/\alpha] \circ \sigma)\theta\rho_2$ (resp. $([\rho'/\beta] \circ \sigma)\theta\rho_3$) is in $|A|_\varphi$.

At last, if the proof-term is reduced by the ultra-reduction rule then the reduct is $\sigma\theta\rho_2$ or $\sigma\theta\rho_3$ and these proof-terms are in $|A|_\varphi$.

Hence, the proof-term $\sigma\theta(\delta\ \rho_1\ \alpha\rho_2\ \beta\rho_3)$ is an element of $|A|_\varphi$.

3.4.10. $\bot$-*elim.* The proof-term π has the form $(\delta_\bot\ \rho)$ with ρ being a proof-term of $\bot$. We have $\sigma\theta\pi = (\delta_\bot\ \sigma\theta\rho)$. By induction hypothesis, the proof-term $\sigma\theta\rho$ is an element of $|\bot|_\varphi$. Hence, it is strongly normalizable. Let n be the maximum length of reduction sequences issued from this proof-term. We prove by induction on n that $(\delta_\bot\ \sigma\theta\rho)$ is in $|A|_\varphi$. Since this proof-term is neutral, we only need to prove that every of its one step reducts is in $|A|_\varphi$. The reduction can only take place in $\sigma\theta\rho$ and we apply the induction hypothesis.

Hence, the proof-term $\sigma\theta(\delta_\bot\ \rho)$ is an element of $|A|_\varphi$.

3.4.11. $\forall$-*elim.* The proof-term π has the form $(\rho\ t)$ where ρ is a proof of some proposition $\forall x\ B$ and $A = [t/x]B$. We have $\sigma\theta\pi = (\sigma\theta\rho\ \theta t)$. By induction hypothesis, the proof-term $\sigma\theta\rho$ is in $|\forall x\ B|_\varphi$. Hence, it is strongly normalizable. Let n be the maximum length of a reduction sequence issued from this proof-term. We prove by induction on n that $(\sigma\theta\rho\ \theta t)$ is in the set $|A|_\varphi$. As this proof-term is neutral, we only need to prove that every of its one step reducts is in $|A|_\varphi$. If the reduction takes place in $\sigma\theta\rho$ then we apply the induction hypothesis. Otherwise $\sigma\theta\rho$ has the form $\lambda x\ \rho'$ and the reduct is in $[\theta t/x]\rho'$. By the definition of $|\forall x\ A|_\varphi$ this proof-term is in $|A|_\varphi$.

Hence, the proof-term $\sigma\theta(\rho\ t)$ is an element of $|A|_\varphi$.

3.4.12. $\exists$-*elim.* The proof-term π has the form $(\delta_\exists\ \rho_1\ \alpha x\rho_2)$ where ρ_1 is a proof of some proposition $\exists x\ B$ and ρ_2 is a proof of A. We have $\sigma\theta\pi = (\delta_\exists\ \sigma\theta\rho_1\ \alpha x\sigma\theta\rho_2)$. By induction hypothesis, the proof-term $\sigma\theta\rho_1$ is in the set $|\exists x\ B|_\varphi$ and the proof-term $\sigma\theta\rho_2$ is in the set $|A|_\varphi$. Hence, these proof-terms are strongly normalizable. Let n and n' be the maximum length of reduction sequences issued from these proof-terms. We prove by induction on $n + n'$ that $(\delta_\exists\ \sigma\theta\rho_1\ x\alpha\sigma\theta\rho_2)$ is in $|A|_\varphi$. As this proof-term is neutral we only need to prove that every of its one step reducts is in $|A|_\varphi$. If the reduction takes place in $\sigma\theta\rho_1$ or $\sigma\theta\rho_2$ then we apply the induction hypothesis.

Otherwise, if $\sigma\theta\rho_1$ has the form $\langle t, \rho'\rangle$ and the reduct is $[\rho'/\alpha][t/x]\sigma\theta\rho_2 = ([\rho'/\alpha] \circ [t/x]\sigma)([t/x] \circ \theta)\rho_2$. By the definition of $|\exists x\ A|_\varphi$, there exists an element E of such that the proof-term ρ' is in $|B|_{\varphi+\langle x,E\rangle}$. Thus, by induction hypothesis, the proof-term $([\rho'/\alpha] \circ [t/x]\sigma)([t/x] \circ \theta)\rho_2$ is in $|A|_{\varphi+\langle x,E\rangle}$, *i.e.* in $|A|_\varphi$.

At last, if the reduction rule is a ultra-reduction rule, then the proof-term reduces to $\sigma\theta\rho_2$ that is in $|B|_\varphi$.

Hence, the proof-term $\sigma\theta(\delta_\exists\ \rho_1\ \alpha x\rho_2)$ is an element of $|A|_\varphi$. ⊣

COROLLARY 3.1. *If a congruence $\equiv$ has a pre-model, then every proof modulo $\equiv$ is strongly normalizable.*

PROOF. Every proof of A is in $|A|_\emptyset$ and hence strongly normalizable. ⊣

3.5. Pre-model construction. Constructing the pre-model for a given theory, is the part of the consistency proof that bears the logical complexity, *i.e.* it is the part of the proof that cannot be done in the theory itself. The construction for simple type theory follows essentially the original proof of Girard [16, 17]. The others are more typical of deduction modulo.

3.5.1. *Simple type theory.*

PROPOSITION 3.3. *Simple type theory has a pre-model, hence proof-terms normalize in simple type theory.*

PROOF. We construct a pre-model as follows. The essential point is that we anticipate the fact that objects of sort o actually represent propositions, by interpreting them as reducibility candidates.

$$\begin{aligned}
\mathcal{M}_\iota &= \{0\}\\
\mathcal{M}_o &= \mathcal{C}\\
\mathcal{M}_{T\to U} &= \mathcal{M}_U^{\mathcal{M}_T}
\end{aligned}$$

$$\begin{aligned}
\hat{S}_{T,U,V} &= a \mapsto (b \mapsto (c \mapsto a(c)(b(c))))\\
\hat{K}_{T,U} &= a \mapsto (b \mapsto a)\\
\hat{\alpha}(a,b) &= a(b)\\
\hat{\varepsilon}(a) &= a
\end{aligned}$$

$$\begin{aligned}
\hat{\Rightarrow}(a,b) &= \{\pi \in \mathcal{SN} | \pi \rhd \lambda\alpha\ \pi_1 \Rightarrow \forall \pi' \in a.[\pi'/\alpha]\pi_1 \in b\}\\
\hat{\wedge}(a,b) &= \{\pi \in \mathcal{SN} | \pi \rhd \langle \pi_1, \pi_2\rangle \Rightarrow \pi_1 \in a \wedge \pi_2 \in b\}\\
\hat{\vee}(a,b) &= \{\pi \in \mathcal{SN} | (\pi \rhd i(\pi_1) \Rightarrow \pi_1 \in a) \wedge (\pi \rhd i(\pi_2) \Rightarrow \pi_2 \in b)\}\\
\hat{\bot} &= \mathcal{SN}\\
\hat{\forall}_T(a) &= \{\pi \in \mathcal{SN} | \pi \rhd \lambda x\ \pi_1 \Rightarrow \forall t : T.\forall E \in \mathcal{M}_T.[t/x]\pi_1 \in a(E)\}\\
\hat{\exists}_T(a) &= \{\pi \in \mathcal{SN} | \pi \rhd \langle t, \pi_2\rangle \Rightarrow \exists E \in \mathcal{M}_T.\pi_2 \in a(E)\}
\end{aligned}$$

We do not detail the proof that if $A \equiv B$ then $|A|_\varphi = |B|_\varphi$. ⊣

In section 4.2, we study cut elimination for classical sequent calculus. This is done using a, so-called, $\neg\neg$-translation. To this matter, we need the following proposition.

PROPOSITION 3.4. *The variant of simple type theory with the following rewrite system has a pre-model*

$$\begin{aligned}
\alpha(\alpha(\alpha(S_{T,U,V},x),y),z) &\to \alpha(\alpha(x,z),\alpha(y,z))\\
\alpha(\alpha(K_{T,U},x),y) &\to x\\
\varepsilon(\alpha(\alpha(\dot\Rightarrow,x),y)) &\to \neg\neg\varepsilon(x) \Rightarrow \neg\neg\varepsilon(y)\\
\varepsilon(\alpha(\alpha(\dot\wedge,x),y)) &\to \neg\neg\varepsilon(x) \wedge \neg\neg\varepsilon(y)\\
\varepsilon(\alpha(\alpha(\dot\vee,x),y)) &\to \neg\neg\varepsilon(x) \vee \neg\neg\varepsilon(y)\\
\varepsilon(\dot\bot) &\to \bot\\
\varepsilon(\alpha(\dot\forall,x)) &\to \forall y\ \neg\neg\varepsilon(\alpha(x,y))\\
\varepsilon(\alpha(\dot\exists,x)) &\to \exists y\ \neg\neg\varepsilon(\alpha(x,y))
\end{aligned}$$

where $\neg A$ is a notation for $A \Rightarrow \bot$.

PROOF. We take the same pre-model as above, but for the interpretation of the symbols $\dot\Rightarrow$, $\dot\wedge$, $\dot\vee$, $\dot\bot$, $\dot\forall$ and $\dot\exists$. We first define the following functions.

$$\begin{aligned}
\tilde\Rightarrow(a,b) &= \{\pi \in \mathcal{SN} | \pi \rhd \lambda\alpha\ \pi_1 \Rightarrow \forall \pi' \in a.[\pi'/\alpha]\pi_1 \in b\}\\
\tilde\wedge(a,b) &= \{\pi \in \mathcal{SN} | \pi \rhd \langle \pi_1, \pi_2\rangle \Rightarrow \pi_1 \in a \wedge \pi_2 \in b\}\\
\tilde\vee(a,b) &= \{\pi \in \mathcal{SN} | (\pi \rhd i(\pi_1) \Rightarrow \pi_1 \in a) \wedge (\pi \rhd i(\pi_2) \Rightarrow \pi_2 \in b)\}\\
\tilde\bot &= \mathcal{SN}\\
\tilde\forall_T(a) &= \{\pi \in \mathcal{SN} | \pi \rhd \lambda x\ \pi_1 \Rightarrow \forall t : T. \forall E \in \mathcal{M}_T.[t/x]\pi_1 \in a(E)\}\\
\tilde\exists_T(a) &= \{\pi \in \mathcal{SN} | \pi \rhd \langle t, \pi_2\rangle \Rightarrow \exists E \in \mathcal{M}_T.\pi_2 \in a(E)\}\\
\tilde\neg(a) &= \tilde\Rightarrow(a, \tilde\bot)
\end{aligned}$$

Then we take

$$\begin{aligned}
\hat\Rightarrow(a,b) &= \tilde\Rightarrow(\tilde\neg(\tilde\neg(a)), \tilde\neg(\tilde\neg(b)))\\
\hat\wedge(a,b) &= \tilde\wedge(\tilde\neg(\tilde\neg(a)), \tilde\neg(\tilde\neg(b)))\\
\hat\vee(a,b) &= \tilde\vee(\tilde\neg(\tilde\neg(a)), \tilde\neg(\tilde\neg(b)))\\
\hat\bot &= \tilde\bot\\
\hat\forall_T(a) &= \tilde\forall(x \mapsto \tilde\neg(\tilde\neg(a(x))))\\
\hat\exists_T(a) &= \tilde\exists(x \mapsto \tilde\neg(\tilde\neg(a(x))))
\end{aligned}$$

⊣

3.5.2. *Quantifier free rewrite systems.* In the case of simple type theory, it is the presence of the quantifier $\forall$ on the right hand part of one of the rewrite schemes that is responsible for the impredicativity of the resulting theory. Quantifier free rewrite systems define predicative theories and have a generic and predicative proof of cut elimination.

DEFINITION 3.7 (Quantifier free). *A rewrite system is said to be quantifier free if no quantifier appears on the right hand side of any of its rules.*

PROPOSITION 3.5. *A quantifier free confluent terminating rewrite systems has a pre-model, hence proof-terms normalize in deduction modulo such a rewrite system.*

PROOF. By induction over proposition structure, we associate a set of proof-terms to each each normal closed quantifier free proposition.

$$\begin{array}{rcl}
\Psi(A) & = & \mathcal{SN} \text{ if } A \text{ is atomic} \\
\Psi(A \Rightarrow B) & = & \{\pi \in \mathcal{SN} | \pi \rhd \lambda\alpha\ \pi_1 \Rightarrow \forall \pi' \in \Psi(A).[\pi'/\alpha]\pi_1 \in \Psi(B)\} \\
\Psi(A \wedge B) & = & \{\pi \in \mathcal{SN} | \pi \rhd (\pi_1, \pi_2) \Rightarrow \pi_1 \in \Psi(A) \wedge \pi_2 \in \Psi(B)\} \\
\Psi(A \vee B) & = & \{\pi \in \mathcal{SN} | \pi \rhd i(\pi_1) \Rightarrow \pi_1 \in \Psi(A) \wedge \pi \rhd i(\pi_2) \Rightarrow \pi_2 \in \Psi(B)\} \\
\Psi(\bot) & = & \mathcal{SN}
\end{array}$$

We define a pre-model as follows.

Let $\mathcal{M}_T$ be the set of normal closed terms of sort T.

$$\hat{f}(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)\downarrow$$
$$\hat{P}(t_1, \ldots, t_n) = \Psi((P(t_1, \ldots, t_n))\downarrow).$$

where $A\downarrow$ (resp. $t\downarrow$) is the normal form of the proposition A (resp. term t).

We prove, by an easy induction, that $|A|_\varphi = |B|_\varphi$ when $A \equiv B$. ⊣

REMARK 3.1. *In this normalization proof we use the fact that some sets are reducibility candidates, but we never quantify over all reducibility candidates, reflecting the fact that we here deal with predicative systems.*

3.5.3. *Positive rewrite systems.* Finally, for some rewrite systems, pre-models can be constructed by a fixed point construction.

DEFINITION 3.8. *A rewrite system is said to be* positive *if it rewrites atomic propositions to propositions containing only positive occurrences of atomic propositions.*

DEFINITION 3.9. *A pre-model is said to be* syntactical *if*

- $\mathcal{M}_T = \mathcal{T}_T/\equiv$ *where* $\mathcal{T}_T$ *is the set of closed terms of sort* T*,*
- *If* f *is a function symbol,* $\hat{f}$ *is the function that maps the classes* $e_1, ..., e_n$ *to the class of the term* $f(t_1, \ldots, t_n)$ *where* $t_1, ..., t_n$ *are elements of* $e_1, ..., e_n$ *(since the relation* $\equiv$ *is a congruence, this does not depend of the choice of representatives).*

A syntactical pre-model is defined solely by the interpretation of predicate symbols.

DEFINITION 3.10. *Let* $\mathcal{M}_1$ *and* $\mathcal{M}_2$ *be two syntactical pre-models. We write* $\hat{P}_1$ *for the denotation of* P *in* $\mathcal{M}_1$ *and* $\hat{P}_2$ *for the denotation of* P *in* $\mathcal{M}_2$

We say that $\mathcal{M}_1 \leq \mathcal{M}_2$ *if and only if for any predicate symbol* P *and closed terms* $t_1, \ldots, t_n$ *we have*

$$\hat{P}_1(t_1, \ldots, t_n) \subseteq \hat{P}_2(t_1, \ldots, t_n)$$

The set of syntactical pre-models is a complete lattice for the order $\leq$.

PROPOSITION 3.6. *Let* $\mathcal{R}$ *be a confluent and normalizing rewrite system. If the system* $\mathcal{R}$ *is positive then it has a pre-model, hence proof-terms normalize modulo* $\mathcal{R}$*.*

PROOF. Let $\mathcal{F}$ be the function mapping syntactical pre-models to syntactical pre-models defined by

$$\mathcal{F}(\mathcal{M})(P)(t_1,\dots,t_n) = |P(t_1,\dots,t_n)\downarrow|_{\mathcal{M},\emptyset}.$$

As the system $\mathcal{R}$ is positive the function $\mathcal{F}$ is monotone. Hence, as the set of syntactical pre-models is a complete lattice, it has a fixed point. This fixed point is a pre-model of the rewrite system. ⊣

This construction also applies for a non terminating rewrite system, provided it is deterministic.

PROPOSITION 3.7. *Let $\mathcal{R}$ be a rewrite system such that any atomic proposition has at most one one-step reduct. If the system $\mathcal{R}$ is positive then it has a pre-model, hence proof-terms normalize modulo $\mathcal{R}$.*

PROOF. Let $\mathcal{F}$ be the function mapping syntactical pre-models to syntactical pre-models defined by

$$\mathcal{F}(\mathcal{M})(P)(t_1,\dots,t_n) = |P(t_1,\dots,t_n)+|_{\mathcal{M},\emptyset}$$

where $A+$ is the unique one-step reduct of A if it exists and A otherwise. Again, since the system $\mathcal{R}$ is positive the function $\mathcal{F}$ is monotone and again, since the set of syntactical pre-models is a complete lattice, it has a fixed point. This fixed point is a pre-model of the rewrite system. ⊣

3.5.4. *The term case.*

PROPOSITION 3.8. *A a congruence induced by a congruence on terms has a pre-model and hence proof-terms normalize modulo this congruence.*

PROOF. Take $\mathcal{SN}$ for all atomic propositions, *i.e.* for all P, $\hat{P}$ is the constant function equal to $\mathcal{SN}$. ⊣

3.6. Normalization with commutative cuts. In order to lift the cut elimination theorem from natural deduction to sequent calculus we need a cut elimination theorem, taking into account the so-called *commutative cuts*.

A usual cut is an introduction rule immediately followed by an elimination rule on the same symbol. A commutative cut is a introduction followed by an elimination, but separated by a sequence of eliminations of symbols $\vee$, $\bot$ or $\exists$. When a proof-term contains a commutative cut, it is possible to permute the eliminations until the introduction and the elimination form a usual cut that then can be reduced. Typically, we want a rule transforming the proof

$$\dfrac{\dfrac{\Gamma\vdash_{\equiv} A\vee B \qquad \Gamma,A\vdash_{\equiv} C\Rightarrow D \qquad \Gamma,B\vdash_{\equiv} C\Rightarrow D}{\Gamma\vdash_{\equiv} C\Rightarrow D}\vee\text{-elim} \qquad \Gamma\vdash_{\equiv} C}{\Gamma\vdash_{\equiv} D}\Rightarrow\text{-elim}$$

into the proof

$$\dfrac{\Gamma\vdash_{\equiv} A\vee B \qquad \dfrac{\Gamma,A\vdash_{\equiv} C\Rightarrow D\ \Gamma,A\vdash_{\equiv} C}{\Gamma,A\vdash_{\equiv} D}\Rightarrow\text{-elim} \qquad \dfrac{\Gamma,B\vdash_{\equiv} C\Rightarrow D\ \Gamma,B\vdash_{\equiv} C}{\Gamma,B\vdash_{\equiv} D}\Rightarrow\text{-elim}}{\Gamma\vdash_{\equiv} D}\vee\text{-elim}$$

Which, using the notation of proofs as terms, corresponds to the additional rule

$$((\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3)\ \pi) \hookrightarrow (\delta\ \pi_1\ \alpha(\pi_2\ \pi)\ \beta(\pi_3\ \pi))$$

More generally reductions with commutative cuts are obtained by adding more rules to those of figure 4 yielding the rules of figure 6. We write $\hookrightarrow$ for reduction with commutative cuts. Notice that, unlike $\triangleright$, the relation $\hookrightarrow$ enjoys subject reduction.

We now want to prove that each proof has a normal form for this rewrite system.

DEFINITION 3.11 (Elimination context). *An elimination context is a proof-term of the form*

$$E := () \mid (E\ \pi) \mid \mathit{fst}(E) \mid \mathit{snd}(E) \mid (\delta\ E\ \pi_1\ \pi_2) \mid (\delta_\bot\ E) \mid (E\ t) \mid (\delta_\exists\ E\ x\alpha\pi)$$

where $()$ *is a distinguished variable.*

We write $E(\pi)$ *for the proof-term* $[\pi/()]E$.

DEFINITION 3.12 (Simple proof-term). *A simple proof-term is a proof-term of the form*

$$S := \alpha \mid (S\ \pi) \mid \mathit{fst}(S) \mid \mathit{snd}(S) \mid (S\ t)$$

PROPOSITION 3.9. *Every* $\rightarrow$*-normal proof-term* π *is either an introduction, a simple proof-term or has the form* $E(\delta\ S\ \alpha\pi_1\ \beta\pi_2)$, $E(\delta_\exists\ S\ x\alpha\pi_1)$ *or* $E(\delta_\bot\ S)$.

PROOF. By induction over the structure of the proof-term π.

- If π is a variable then it is simple.
- If π is an introduction then it is an introduction.
- If π is an elimination, then it has the form $(\rho\ \rho_1)$, $\mathit{fst}(\rho)$, $\mathit{snd}(\rho)$, $(\delta\ \rho\ \alpha\rho_1\ \beta\rho_2)$, $(\delta_\bot\ \rho)$, $(\rho\ t)$ or $(\delta_\exists\ \rho\ x\alpha\rho_1)$.

  In all these cases, by induction hypothesis the proof-term ρ is either an introduction, a simple proof-term or has the form $E(\delta\ S\ \alpha\pi_1\ \beta\pi_2)$, $E(\delta_\exists\ S\ x\alpha\pi_1)$ or $E(\delta_\bot\ S)$.

  Since π is normal, ρ is not an introduction.

  If ρ is simple then π is either simple or has the form $(\delta\ S\ \alpha\rho_1\ \beta\rho_2)$, $(\delta_\bot\ S)$, $(\delta_\exists\ S\ x\alpha\rho_1)$.

  If ρ has the form $E(\delta\ S\ \alpha\pi_1\ \beta\pi_2)$, $E(\delta_\exists\ S\ x\alpha\pi_1)$ or $E(\delta_\bot\ S)$, then π has the form $E'(\delta\ S\ \alpha\pi_1\ \beta\pi_2)$, $E'(\delta_\exists\ S\ x\alpha\pi_1)$ or $E'(\delta_\bot\ S)$.

$\dashv$

PROPOSITION 3.10. *A simple proof-term can reduce to simple proof-terms only.*

PROPOSITION 3.11 (Weak normalization). *Every proof has a* $\hookrightarrow$*-normal form.*

PROOF. Let π be a proof-term, we write $n(\pi)$ for the length of the longest $\triangleright$-reduction in π and $p(\pi)$ for the size of π. By induction over the lexicographic ordering $\langle n(\pi), p(\pi)\rangle$, we prove that every proof-term π has a $\hookrightarrow$-normal form.

- If the proof-term π is not $\rightarrow$-normal then it $\rightarrow$-reduces to some proof-term π'. We have $n(\pi') < n(\pi)$. By the induction hypothesis π' has a $\hookrightarrow$-normal form. Hence π has a $\hookrightarrow$-normal form.

$$(\lambda\alpha\ \pi_1\ \pi_2) \hookrightarrow [\pi_2/\alpha]\pi_1$$
$$\mathit{fst}(\langle\pi_1,\pi_2\rangle) \hookrightarrow \pi_1$$
$$\mathit{snd}(\langle\pi_1,\pi_2\rangle) \hookrightarrow \pi_2$$
$$(\delta\ i(\pi_1)\ \alpha\pi_2\ \beta\pi_3) \hookrightarrow [\pi_1/\alpha]\pi_2$$
$$(\delta\ j(\pi_1)\ \alpha\pi_2\ \beta\pi_3) \hookrightarrow [\pi_1/\beta]\pi_3$$
$$(\lambda x\ \pi\ t) \hookrightarrow [t/x]\pi$$
$$(\delta_\exists\ \langle t,\pi_1\rangle\ \alpha x\pi_2) \hookrightarrow [t/x,\pi_1/\alpha]\pi_2$$

$$((\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3)\ \pi) \hookrightarrow (\delta\ \pi_1\ \alpha(\pi_2\ \pi)\ \beta(\pi_3\ \pi))$$
$$\mathit{fst}(\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3) \hookrightarrow (\delta\ \pi_1\ \alpha\mathit{fst}(\pi_2)\ \beta\mathit{fst}(\pi_3))$$
$$\mathit{snd}(\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3) \hookrightarrow (\delta\ (\pi_1\ \alpha\mathit{snd}(\pi_2)\ \beta\mathit{snd}(\pi_3)))$$
$$(\delta\ (\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3)\ \alpha'\pi'_1\ \beta'\pi'_2) \hookrightarrow (\delta\ \pi_1\ \alpha(\delta\ \pi_2\ \alpha'\pi'_1\ \beta'\pi'_2)\ \beta(\delta\ \pi_3\ \alpha'\pi'_1\ \beta'\pi'_2))$$
$$(\delta_\bot\ (\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3)) \hookrightarrow (\delta\ \pi_1\ \alpha(\delta_\bot\ \pi_2)\ \beta(\delta_\bot\ \pi_3))$$
$$((\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3)\ t) \hookrightarrow ((\delta\ \pi_1\ \alpha(\pi_2\ t)\ \beta(\pi_3\ t)))$$
$$(\delta_\exists\ (\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3)\ x\alpha'\pi) \hookrightarrow (\delta\ \pi_1\ \alpha(\delta_\exists\ \pi_2\ x\alpha'\pi)\ \beta(\delta_\exists\ \pi_3\ x\gamma\pi))$$

$$((\delta_\exists\ \pi_1\ x\alpha\pi_2)\ \pi) \hookrightarrow (\delta_\exists\ \pi_1\ \alpha(\pi_2\ \pi))$$
$$\mathit{fst}(\delta_\exists\ \pi_1\ x\alpha\pi_2) \hookrightarrow (\delta_\exists\ \pi_1\ x\alpha\mathit{fst}(\pi_2))$$
$$\mathit{snd}(\delta_\exists\ \pi_1\ x\alpha\pi_2) \hookrightarrow (\delta_\exists\ \pi_1\ x\alpha\mathit{snd}(\pi_2))$$
$$(\delta\ (\delta_\exists\ \pi_1\ x\alpha\pi_2)\ \alpha'\pi'_1\ \beta'\pi'_2) \hookrightarrow (\delta_\exists\ \pi_1\ x\alpha(\delta\ \pi_2\ \alpha'\pi'_1\ \beta'\pi'_2))$$
$$(\delta_\bot\ (\delta_\exists\ \pi_1\ x\alpha\pi_2)) \hookrightarrow (\delta_\exists\ \pi_1\ x\alpha(\delta_\bot\ \pi_2))$$
$$((\delta_\exists\ \pi_1\ x\alpha\pi_2)\ t) \hookrightarrow (\delta_\exists\ \pi_1\ x\alpha(\pi_2\ t))$$
$$(\delta_\exists\ (\delta_\exists\ \pi_1\ x\alpha\pi_2)\ x'\alpha'\pi) \hookrightarrow (\delta_\exists\ \pi_1\ x\alpha(\delta_\exists\ \pi_2\ x'\alpha'\pi))$$

$$((\delta_\bot\ \pi_1)\ \pi) \hookrightarrow (\delta_\bot\ \pi_1)$$
$$\mathit{fst}(\delta_\bot\ \pi_1) \hookrightarrow (\delta_\bot\ \pi_1)$$
$$\mathit{snd}(\delta_\bot\ \pi_1) \hookrightarrow (\delta_\bot\ \pi_1)$$
$$(\delta\ (\delta_\bot\ \pi_1)\ \alpha'\pi'_1\ \beta'\pi'_2) \hookrightarrow (\delta_\bot\ \pi_1)$$
$$(\delta_\bot\ (\delta_\bot\ \pi_1)) \hookrightarrow (\delta_\bot\ \pi_1)$$
$$((\delta_\bot\ \pi_1)\ t) \hookrightarrow (\delta_\bot\ \pi_1)$$
$$(\delta_\exists\ (\delta_\bot\ \pi_1)\ x\alpha'\pi) \hookrightarrow (\delta_\bot\ \pi_1)$$

FIGURE 6. Reduction rules with commutative cuts

- If the proof-term π is $\to$-normal then, using proposition 3.9, it is either an introduction, a simple proof-term or a proof-term of the form $E(\delta\ S\ \alpha\pi_1\ \beta\pi_2)$, $E(\delta_\exists\ S\ x\alpha\pi_1)$ or $E(\delta_\bot\ S)$.
  - If π is an introduction, it has the form $\lambda\alpha\ \pi_1$, $\langle\pi_1,\pi_2\rangle$, $i(\pi_1)$, $j(\pi_1)$, $\lambda x\ \pi_1$, $\langle t,\pi_1\rangle$.
    In all theses cases, we have $n(\pi_i) \leq n(\pi)$ and $p(\pi_i) < p(\pi)$ hence by induction hypothesis the proof-terms π_i have a $\hookrightarrow$-normal forms π'_i. The proof-term $\lambda\alpha\ \pi'_1$, (resp. $\langle\pi'_1,\pi'_2\rangle$, $i(\pi'_1)$, $j(\pi'_1)$, $\lambda x\ \pi'_1$ or $\langle t,\pi'_1\rangle$) is a $\hookrightarrow$-normal form of π.

– If π is simple then we prove by structural induction that it has a normal form.
  * If it is a variable then it is its own normal form.
  * If it has the form $(S\ \pi_1)$ then by induction hypothesis S has a normal form S'. This proof-term is simple. We have $n(\pi_1) \leq n(\pi)$ and $p(\pi_1) < p(\pi)$, hence, by induction hypothesis, π_1 has a normal form π'_1. The proof-term $(S'\ \pi'_1)$ is a normal form of π.
  * If π has the form $\mathit{fst}(S)$ or $\mathit{snd}(S)$ then by induction hypothesis the proof-term S has a normal form S'. This proof-term is simple. The proof-term $\mathit{fst}(S')$ or $\mathit{snd}(S')$ is a normal form of π.
  * If π has the form $(S\ t)$ then by induction hypothesis S has a normal form S'. This proof-term is simple. The proof-term $(S'\ t)$ is a normal form of π.

– If π has the form $E(\delta\ S\ \alpha\pi_1\ \beta\pi_2)$, $E(\delta_\exists\ S\ x\alpha\pi_1)$ or $E(\delta_\bot\ S)$, then it $\hookrightarrow$-reduces to $(\delta\ S\ \alpha E(\pi_1)\ \beta E(\pi_2))$, $(\delta_\exists\ S\ x\alpha E(\pi_1))$ or $(\delta_\bot\ S)$.
  We have $n(S) \leq n(\pi)$ and $p(S) < p(\pi)$ hence, by induction hypothesis the proof-term S has a normal form S'. This proof-term is simple.
  We have $\pi \rhd^+ E(\pi_i)$ (recall that $\rhd$ stands for ultra-reduction). Hence $n(E(\pi_i)) < n(\pi)$. Thus, by induction hypothesis the proof-term $E(\pi_i)$ has a $\hookrightarrow$-normal form π'_i.
  The proof-term $(\delta\ S'\ \alpha\pi'_1\ \beta\pi'_2)$, $(\delta_\exists\ S'\ x\alpha\pi'_1)$ or $(\delta_\bot\ S')$ is a normal form of π.

$\dashv$

§4. Cut elimination in sequent calculus modulo.

4.1. Cut elimination for the intuitionistic sequent calculus modulo.

Following a usual proof, we show that $\hookrightarrow$-normal proof-terms in natural deduction can be translated as cut free proofs in sequent calculus.

REMARK 4.1. *In natural deduction, the context of the main premise of an elimination is the same as that of the conclusion.*

PROPOSITION 4.1. *If a sequent $\Gamma \vdash_\equiv P$ has a $\hookrightarrow$-normal proof in intuitionistic natural deduction modulo, then it has a cut free proof in intuitionistic sequent calculus modulo.*

PROOF. By induction on the size of the normal proof-term of $\Gamma \vdash_\equiv P$

- If the last rule is an axiom then the result is obvious.
- If the last rule is an introduction rule, we apply the induction hypothesis, to the subproofs and we use the corresponding right rule.
- If the last rule is an elimination rule, then the proof ends with a sequence of elimination rules on the main premise and we consider the first rule that is not an elimination. This rule cannot be a introduction rule because the proof is normal, thus it is an axiom. We focus on the first rule after this axiom.

– If this rule is a $\Rightarrow$-elim rule. The proof has the shape

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash_{\equiv} C}\,\text{axiom} \quad \dfrac{\pi_1}{\Gamma \vdash_{\equiv} A}}{\dfrac{\Gamma \vdash_{\equiv} B}{\dfrac{\pi_2}{\Gamma \vdash_{\equiv} P}}}\,\Rightarrow\text{-elim}$$

where $C \equiv A \Rightarrow B$.
Adding the proposition B in every context of π_2 we get a proof of $\Gamma, B \vdash_{\equiv} P$ under the assumption $\Gamma, B \vdash_{\equiv} B$, we can close this proof with an axiom and the proof π_2' obtained this way is shorter than the proof we started with.
We apply the induction hypothesis to π_1 and π_2' yielding cut free proofs ρ_1 and ρ_2 in sequent calculus of $\Gamma \vdash_{\equiv} A$ and $\Gamma, B \vdash_{\equiv} P$.
The context Γ contains the proposition C. We build the proof

$$\dfrac{\dfrac{\dfrac{\rho_1}{\Gamma \vdash_{\equiv} A} \quad \dfrac{\rho_2}{\Gamma, B \vdash_{\equiv} P}}{\Gamma, C \vdash_{\equiv} P}\,\Rightarrow\text{-left}}{\Gamma \vdash_{\equiv} P}\,\text{contraction}$$

– If this rule is a $\wedge$-elim rule. The proof has the shape

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash_{\equiv} C}\,\text{axiom}}{\Gamma \vdash_{\equiv} A}\,\wedge\text{-elim}}{\dfrac{\pi}{\Gamma \vdash_{\equiv} P}}}$$

where $C \equiv A \wedge B$.
Adding the propositions A and B in every context of π we get a proof of $\Gamma, A, B \vdash_{\equiv} P$ under the assumption $\Gamma, A, B \vdash_{\equiv} A$, we can close this proof with an axiom and the proof π' obtained this way is shorter than the proof we started with.
We apply the induction hypothesis to π' yielding a cut free proof ρ in sequent calculus of $\Gamma, A, B \vdash_{\equiv} P$
The context Γ contains the proposition C. We build the proof

$$\dfrac{\dfrac{\dfrac{\rho}{\Gamma, A, B \vdash_{\equiv} P}}{\Gamma, C \vdash_{\equiv} P}\,\wedge\text{-left}}{\Gamma \vdash_{\equiv} P}\,\text{contraction}$$

– If this rule is a $\vee$-elim rule. The proof has the shape

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash_{\equiv} D}\,\text{axiom} \quad \dfrac{\pi_1}{\Gamma, A \vdash_{\equiv} C} \quad \dfrac{\pi_2}{\Gamma, B \vdash_{\equiv} C}}{\Gamma \vdash_{\equiv} C}\,\vee\text{-elim}}{\dfrac{\pi_3}{\Gamma \vdash_{\equiv} P}}$$

where $D \equiv A \vee B$.

As the proof is $\hookrightarrow$-normal and π_3 contains only eliminations, π_3 is empty and the proof has the shape

$$\dfrac{\dfrac{}{\Gamma \vdash_{\equiv} D}\,\text{axiom} \quad \dfrac{\overset{\pi_1}{\Gamma, A \vdash_{\equiv} P} \quad \dfrac{\overset{\pi_2}{\Gamma, B \vdash_{\equiv} P}}{}}{}}{\Gamma \vdash_{\equiv} P}\,\vee\text{-elim}$$

We apply the induction hypothesis to π_1 and π_2 yielding cut free proofs ρ_1 and ρ_2 in sequent calculus of $\Gamma, A \vdash_{\equiv} P$ and $\Gamma, B \vdash_{\equiv} P$.
The context Γ contains the proposition D. We build the proof

$$\dfrac{\dfrac{\dfrac{\overset{\rho_1}{\Gamma, A \vdash_{\equiv} P} \quad \dfrac{\overset{\rho_2}{\Gamma, B \vdash_{\equiv} P}}{}}{\Gamma, D \vdash_{\equiv} P}\,\Rightarrow\text{-left}}{\Gamma \vdash_{\equiv} P}\,\text{contraction}}{}$$

– If this rule is a $\bot$-elim rule. The proof has the shape

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash_{\equiv} B}\,\text{axiom}}{\Gamma \vdash_{\equiv} A}\,\bot\text{-elim}}{\Gamma \vdash_{\equiv} P}\,\pi$$

where $B \equiv \bot$.
As the proof is $\hookrightarrow$-normal and π contains only eliminations, π is empty and the proof has the shape

$$\dfrac{\dfrac{}{\Gamma \vdash_{\equiv} B}\,\text{axiom}}{\Gamma \vdash_{\equiv} P}\,\bot\text{-elim}$$

The context Γ contains the proposition B. We build the proof

$$\dfrac{\dfrac{}{\Gamma, B \vdash_{\equiv} P}\,\bot\text{-left}}{\Gamma \vdash_{\equiv} P}\,\text{contraction}$$

– If this rule is a $\forall$-elim rule. The proof has the shape

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash_{\equiv} B}\,\text{axiom}}{\Gamma \vdash_{\equiv} C}\,\langle x, A, t\rangle\forall\text{-elim}}{\Gamma \vdash_{\equiv} P}\,\pi$$

where $B \equiv \forall x\ A$ and $C \equiv [t/x]A$.
Adding the proposition C in every context of π we get a proof of $\Gamma, C \vdash_{\equiv} P$ under the assumption $\Gamma, C \vdash_{\equiv} C$, we can close this proof with an axiom and the proof π' obtained this way is shorter than the proof we started with.
We apply the induction hypothesis to π' and yielding a cut free proof ρ in sequent calculus of $\Gamma, C \vdash_{\equiv} P$.

The context Γ contains the proposition B. We build the proof

$$\dfrac{\dfrac{\dfrac{\rho}{\Gamma, C \vdash_{\equiv} P}}{\Gamma, B \vdash_{\equiv} P}\,\langle x, A, t\rangle\ \forall\text{-left}}{\Gamma \vdash_{\equiv} P}\,\text{contraction}$$

– If this rule is a $\exists$-elim rule. The proof has the shape

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash_{\equiv} C}\,\text{axiom} \qquad \dfrac{\pi_1}{\Gamma, A \vdash_{\equiv} B}}{\Gamma \vdash_{\equiv} B}\,\langle x, A\rangle\ \exists\text{-elim}}{\Gamma \vdash_{\equiv} P}$$

where $C \equiv \exists x A$.
As the proof is $\hookrightarrow$-normal and π_2 contains only eliminations, π_2 is empty and the proof has the shape

$$\dfrac{\dfrac{}{\Gamma \vdash_{\equiv} C}\,\text{axiom} \qquad \dfrac{\pi_1}{\Gamma, A \vdash_{\equiv} P}}{\Gamma \vdash_{\equiv} P}\,\langle x, A\rangle\ \exists\text{-elim}$$

We apply the induction hypothesis to π_1 yielding a cut free proof ρ_1 in sequent calculus of $\Gamma, A \vdash_{\equiv} P$.
The context Γ contains the proposition C. We build the proof

$$\dfrac{\dfrac{\dfrac{\rho_1}{\Gamma, A \vdash_{\equiv} P}}{\Gamma, C \vdash_{\equiv} P}\,\langle x, A\rangle\ \exists\text{-left}}{\Gamma \vdash_{\equiv} P}\,\text{contraction}$$

⊣

Corollary 4.1. *If the congruence $\equiv$ has a pre-model, then the cut rule is redundant in intuitionistic sequent calculus modulo $\equiv$.*

Proof. Consider a proof of a sequent $\Gamma \vdash_{\equiv} P$ in sequent calculus, this proof can easily be translated into natural deduction. The $\hookrightarrow$-normal form of this proof can be translated as a cut free proof of $\Gamma \vdash_{\equiv} P$ in sequent calculus. Hence the sequent $\Gamma \vdash_{\equiv} P$ has a cut free proof in sequent calculus. ⊣

4.2. Cut elimination for the classical sequent calculus modulo. In this section we prove cut elimination for classical sequent calculus. We use a traditional style reduction to intuitionistic cut elimination through a $\neg\neg$-*translation*.

Definition 4.1. *Let A be a proposition, the* double negation *of A is the proposition obtained by adding a double negation before each sub-proposition:*

- $A' = \neg\neg A$ *if A is atomic,*
- $(A \Rightarrow B)' = \neg\neg(A' \Rightarrow B')$,
- $(A \wedge B)' = \neg\neg(A' \wedge B')$,
- $(A \vee B)' = \neg\neg(A' \vee B')$,
- $\bot' = \neg\neg\bot$,
- $(\forall x\ A)' = \neg\neg(\forall x\ A')$,

- $(\exists x\ A)' = \neg\neg(\exists x\ A')$.

the light double negation *of A is the proposition obtained by adding a double negation before each sub-proposition except at the root of the proposition:*

- $A'' = A$ *if* A *is atomic,*
- $(A \Rightarrow B)'' = A' \Rightarrow B'$,
- $(A \wedge B)'' = A' \wedge B'$,
- $(A \vee B)'' = A' \vee B'$,
- $\bot'' = \bot$,
- $(\forall x\ A)'' = \forall x\ A'$,
- $(\exists x\ A)'' = \exists x\ A'$.

To a rewrite system $\mathcal{R}\ A_i \to P_i$ we associate the rewrite system $\mathcal{R}'\ A_i \to P_i''$.

PROPOSITION 4.2. *If $A \to_{\mathcal{R}} B$ then $A' \to_{\mathcal{R}'} B'$ and $A'' \to_{\mathcal{R}'} B''$.*

PROOF. By induction on the structure of A. ⊣

COROLLARY 4.2. *If $A \equiv_{\mathcal{R}} B$ then $A' \equiv_{\mathcal{R}'} B'$ and $A'' \equiv_{\mathcal{R}'} B''$.*

PROPOSITION 4.3. *If $\mathcal{R}$ is terminating and confluent, then so is $\mathcal{R}'$.*

PROOF. From a reduction sequence in $\mathcal{R}'$ we can build one of the same length in $\mathcal{R}$. Hence all reduction sequences in $\mathcal{R}'$ are finite and $\mathcal{R}'$ is terminating.

By Newman's theorem, to prove that $\mathcal{R}'$ is confluent, we only need to prove that all critical pairs can be closed. If A is atomic, $A \to^1_{\mathcal{R}'} B$ and $A \to^1_{\mathcal{R}'} C$, then there exists propositions b and c such that $A \to^1_{\mathcal{R}} b$, $A \to^1_{\mathcal{R}} c$, $B = b''$ and $C = c''$. As the system $\mathcal{R}$ is confluent there exists a proposition d such that $b \to_{\mathcal{R}} d$ and $c \to_{\mathcal{R}} d$. We have $B \to_{\mathcal{R}'} d''$ and $C \to_{\mathcal{R}'} d''$. ⊣

PROPOSITION 4.4. *If a sequent $A_1, \ldots, A_n \vdash_{\equiv} B_1, \ldots, B_p$ is provable in classical sequent calculus, then $A'_1, \ldots, A'_n, \neg B''_1, \ldots, \neg B''_p \vdash_{\equiv}$ is provable in intuitionistic sequent calculus.*

PROOF. The proof is an easy induction on the structure of the proof of

$$A_1, \ldots, A_n \vdash_{\equiv} B_1, \ldots, B_p.$$

As an example, let us detail one case. If the last rule is $\vee$-right

$$\dfrac{\dfrac{\pi}{A_1, \ldots, A_n \vdash_{\equiv} C, B_1, \ldots, B_p}}{A_1, \ldots, A_n \vdash_{\equiv} E, B_1, \ldots, B_p}\ \vee\text{-right}$$

Where $E \equiv C \vee D$. By induction hypothesis we have an intuitionistic proof π' of

$$A'_1, \ldots, A'_n, \neg B''_1, \ldots, \neg B''_p, \neg C'' \vdash_{\equiv}$$

We have $E'' \equiv (C \vee D)'' = C' \vee D' = \neg\neg C'' \vee \neg\neg D''$. We build the following proof

$$\dfrac{\dfrac{\dfrac{\dfrac{\pi'}{A'_1, \ldots, A'_n, \neg B''_1, \ldots, \neg B''_p, \neg C'' \vdash_{\equiv}}}{A'_1, \ldots, A'_n, \neg B''_1, \ldots, \neg B''_p \vdash_{\equiv} \neg\neg C''}\ \neg\text{-right}}{A'_1, \ldots, A'_n, \neg B''_1, \ldots, \neg B''_p \vdash_{\equiv} (\neg\neg C'' \vee \neg\neg D'')}\ \vee\text{-right}}{A'_1, \ldots, A'_n, \neg B''_1, \ldots, \neg B''_p, \neg(\neg\neg C'' \vee \neg\neg D'') \vdash_{\equiv}}\ \neg\text{-left}$$

⊣

DEFINITION 4.2. *Consider an intuitionistic sequent* $\Gamma \vdash_{\equiv} \Delta$, i.e. *a sequent such that* Δ *contains at most one proposition. The sequent* $\Gamma \vdash_{\equiv} \Delta$ *is said to represent a classical sequent* $A_1, \ldots, A_n \vdash_{\equiv} B_1, \ldots, B_p$ *if there exists a one-to-one correspondence* ξ *between the formulas of the two sequents such that*

- *if* $\xi(A_i) \in \Gamma$ *then* $\xi(A_i)$ *is either equal to* A_i'' *or* A_i'
- *if* $\xi(A_i) \in \Delta$ *then* $\xi(A_i) = \neg A_i''$
- *if* $\xi(B_i) \in \Delta$ *then* $\xi(B_i)$ *is either equal to* B_i'' *or* B_i'
- *if* $\xi(B_i) \in \Gamma$ *then* $\xi(B_i) = \neg B_i''$.

PROPOSITION 4.5. *If an intuitionistic sequent* $\Gamma \vdash_{\equiv} \Delta$ *represents a classical sequent* $A_1, \ldots, A_n \vdash_{\equiv} B_1, \ldots, B_p$ *and is provable in the cut free intuitionistic sequent calculus modulo* $\mathcal{R}'$, *then the sequent* $A_1, \ldots, A_n \vdash_{\equiv} B_1, \ldots, B_p$ *is provable in the cut free classical sequent calculus modulo* $\mathcal{R}$.

PROOF.

- If the last rule is logical rule applied to a proposition of the form $\neg A_i''$, $\neg B_i''$, A_i' of B_i', then the obtained sequent is also a representation of $A_1, \ldots, A_n \vdash_{\equiv} B_1, \ldots, B_p$ and we can apply the induction hypothesis.
- If the last rule is a logical rule applied to a proposition of the form A_i'' or B_i'', a structural rule or an axiom rule, we apply the same rule to the corresponding proposition in $A_1, \ldots, A_n \vdash_{\equiv} B_1, \ldots, B_p$ and we conclude by the induction hypothesis.

⊣

THEOREM 4.1. *If the rewrite system* $\mathcal{R}'$ *has a pre-model then the cut rule is redundant in the classical sequent calculus modulo* $\mathcal{R}$.

PROOF. If the system $\mathcal{R}'$ has a pre-model then the intuitionistic sequent calculus modulo $\mathcal{R}'$ has the cut elimination property. Let $\Gamma \vdash_{\equiv} \Delta$ be a sequent and $\Gamma' \vdash_{\equiv} \Delta'$ its double negation.

If $\Gamma \vdash_{\equiv} \Delta$ has a proof in the classical sequent calculus modulo $\mathcal{R}$, then $\Gamma' \vdash_{\equiv} \Delta'$ has a proof in the intuitionistic sequent calculus modulo $\mathcal{R}'$, $\Gamma' \vdash_{\equiv} \Delta'$ has a cut free proof in the intuitionistic sequent calculus modulo $\mathcal{R}'$ and $\Gamma \vdash_{\equiv} \Delta$ has a cut free proof in the classical sequent calculus modulo $\mathcal{R}$. ⊣

COROLLARY 4.3.

- *The classical sequent calculus modulo the rules of simple type theory has the cut elimination property.*
- *The classical sequent calculus modulo a confluent and terminating quantifier free rewrite system has the cut elimination property.*
- *The classical sequent calculus modulo any confluent and terminating positive rewrite system has the cut elimination property.*
- *The classical sequent calculus modulo a positive rewrite system such that any atomic proposition has at most one one-step reduct has the cut-elimination property.*
- *The classical sequent calculus modulo a congruence induced by a congruence on terms has the cut-elimination property.*

PROOF. The ¬¬-translation of the rewrite system of simple type theory is that of proposition 3.4. The ¬¬-translation of a quantifier free rewrite system is a quantifier free rewrite system. The ¬¬-translation of a positive rewrite system is a positive rewrite system. The ¬¬-translation of a term rewrite system is a term rewrite system. ⊣

REFERENCES

[1] S.C. BAILIN, *A normalization theorem for set theory*, ***The Journal of Symbolic Logic***, vol. 53 (1988), no. 3, pp. 673–695.

[2] H. BARENDREGT, *Lambda calculi with types*, ***Handbook of logic in computer science*** (S. Abramsky, D.M. Gabbay, and T.S.E. Maibaum, editors), vol. 2, Oxford University Press, 1992.

[3] R.S. BOYER and J.S. MOORE, ***A computational logic***, Academic Press, 1979.

[4] A. CHURCH, *A formulation of the simple theory of types*, ***The Journal of Symbolic Logic***, vol. 5 (1940), no. 1, pp. 56–68.

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

[6] M. CRABBÉ, *Non-normalisation de ZF*, Manuscript, 1974.

[7] ———, *Stratification and cut-elimination*, ***The Journal of Symbolic Logic***, vol. 56 (1991), no. 1, pp. 213–226.

[8] H.B. CURRY, *An analysis of the logical substitution*, ***American Journal of Mathematics***, vol. 51 (1929), pp. 263–384.

[9] G. DOWEK, *Proof normalization for a first-order formulation of higher-order logic*, ***Theorem proving in higher-order logics*** (E.L. Gunter and A. Felty, editors), Lecture Notes in Computer Science, vol. 1275, Springer-Verlag, 1997, pp. 105–119. Rapport de Recherche 3383, *Institut National de Recherche en Informatique et en Automatique*, 1998.

[10] ———, *About folding-unfolding cuts and cuts modulo*, ***Journal of Logic and Computation***, vol. 11 (2001), no. 3, pp. 419–429.

[11] G. DOWEK, TH. HARDIN, and C. KIRCHNER, *Theorem proving modulo*, ***Journal of Automated Reasoning***, (to appear).

[12] J. EKMAN, *Normal proofs in set theory*, Doctoral thesis, Chalmers university of technology and University of Göteborg, 1994.

[13] H.B. ENDERTON, ***A mathematical introduction to logic***, Academic Press, 1972.

[14] M.J. FAY, *First-order unification in an equational theory*, ***Fourth workshop on automated deduction***, 1979, pp. 161–167.

[15] J. GALLIER, ***Logic in computer science***, Harper and Row, 1986.

[16] J.Y. GIRARD, *Interprétation fonctionnelle et élimination des coupures dans l'arithmétique d'ordre supérieur*, Doctoral thesis, Université de Paris 7, 1972.

[17] J.Y. GIRARD, Y. LAFONT, and P. TAYLOR, ***Proofs and types***, Cambridge University Press, 1989.

[18] L. HALLNÄS, *On normalization of proofs in set theory*, Doctoral thesis, University of Stockholm, 1983.

[19] L. HENKIN, *Banishing the rule of substitution for functional variables*, ***The Journal of Symbolic Logic***, vol. 18 (1953), no. 3, pp. 201–208.

[20] J.-M. HULLOT, *Canonical forms and unification*, ***Conference on automated deduction*** (W. Bibel and R. Kowalski, editors), Lecture Notes in Computer Science, vol. 87, Springer-Verlag, 1980, pp. 318–334.

[21] J.-L. KRIVINE and M. PARIGOT, *Programming with proofs*, ***J. Inf. Process. Cybern. EIK***, vol. 26 (1990), no. 3, pp. 149–167.

[22] P. MARTIN-LÖF, ***Intuitionistic type theory***, Bibliopolis, 1984.

[23] R.P. Nederpelt, J.H. Geuvers, and R.C. de Vrijer (editors), ***Selected papers on automath***, Studies in Logic and the Foundations of Mathematics, vol. 133, North-Holland, 1994.

[24] S. Owre and N.Shankar, *The formal semantics of PVS*, ***Technical Report CSL-97-2R***, SRI, 1999.

[25] Ch. Paulin-Mohring, *Inductive definitions in the system Coq - rules and properties*, ***Typed lambda calculi and applications*** (M. Bezem and J.-F. Groote, editors), Lecture Notes in Computer Science, vol. 664, Springer-Verlag, 1993, pp. 328–345.

[26] G. Plotkin, *Building-in equational theories*, ***Machine Intelligence***, vol. 7 (1972), pp. 73–90.

[27] D. Prawitz, ***Natural deduction, a proof-theoretical study***, Almqvist & Wiksell, 1965.

[28] M. Stickel, *Automated deduction by theory resolution*, ***Journal of Automated Reasoning***, vol. 4 (1985), no. 1, pp. 285–289.

[29] W.W. Tait, *Intensional interpretation of functionals of finite type I*, ***The Journal of Symbolic Logic***, vol. 32 (1967), no. 2, pp. 198–212.

[30] ———, *A realizability interpretation of the theory of species*, ***Logic colloquium***, Lecture Notes in Mathematics, vol. 435, 1975, pp. 240–251.

[31] B. Werner, *Une théorie des constructions inductives*, Doctoral thesis, Université Paris 7, 1994.

École polytechnique and INRIA,
LIX, École polytechnique, 91128 Palaiseau cedex, France

E-mail: Gilles.Dowek@polytechnique.fr
URL: http://www.lix.polytechnique.fr/~dowek/

E-mail: Benjamin.Werner@polytechnique.fr
URL: http://www.lix.polytechnique.fr/~werner/