Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Electronic Notes in Theoretical Computer Science 140 (2005) 31–39
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

A Semantics of Realisability for the Classical Propositional Natural Deduction

Karim Nour

Equipe de Logique, Université de Savoie
73376 Le Bourget du Lac, France

Khelifa Saber

Equipe de Logique, Université de Savoie
73376 Le Bourget du Lac, France

Abstract

In this paper, we introduce a semantics of realisability for the classical propositional natural deduction and we prove a correctness theorem. This allows to characterize the operational behaviour of some typed terms.

Keywords: classical natural deduction, semantics of realisability, correctness theorem.

1 Introduction

Natural deduction system is one of the main logical system which was introduced by Gentzen [4] to study the notion of proof. The full classical natural deduction system is well adapted for the human reasoning. By full we mean that all the connectives ($\rightarrow$, $\wedge$ and $\vee$) and $\perp$ (for the absurdity) are considered as primitive and they have their intuitionistic meaning. As usual, the negation is defined by $\neg A = A \rightarrow \perp$. Considering this logic from the computer science of view is interesting because, by the Curry-Howard correspondence,

[1] Email: knour@univ-savoie.fr
[2] Email: ksabe@univ-savoie.fr

doi:10.1016/j.entcs.2005.06.026

formulas can be seen as types for the functional programming languages and correct programs can be extracted. By this correspondence the corresponding calculus is an extension of the $\lambda\mu$-calculus with product and co-product.

Until very recently (see the introduction of [3] for a brief history), no proof of the strong normalization of the cut-elimination procedure was known for full logic. In [3], P. De Groote gives a such proof for classical propositional natural deduction by using the CPS-transformation. R. David and the first author give in [2] a direct and syntactical proof of this result. R. Matthes recently found another semantical proof of this result (see [6]).

In order to prove the strong normalization of classical propositional natural deduction, we introduce in [8] a variant of the reducibility candidates, which was already present in [11]. This method has been introduced by J.Y. Girard. It consists in associating to each type A a set of terms $|A|$, such that every term is in the interpretation of its type (this is called "the adequation lemma"). To the best of our knowledge, we obtain the shortest proof of this result.

In this paper, we define a semantics of realisability of classical propositional natural deduction inspired by [8] and we estabilish a correctness theorem. The idea is to replace the set of strongly normalizing terms used in the proof presented in [8] by a set having the properties necessary to keep the adequation lemma. This result allows to characterize the operational behaviour of terms having some particular types.

The paper is organized as follows. Section 2 is an introduction to the typed system and the relative cut-elimination procedure. In section 3, we define the semantics of realisability and we prove the correctness theorem. In section 4, we give some applications of this result.

2 Notations and definitions

Definition 2.1 We use notations inspired by the paper [1].

(i) Let $\mathcal{X}$ and $\mathcal{A}$ be two disjoint alphabets for distinguishing the λ-variables and μ-variables respectively. We code deductions by using a set of terms $\mathcal{T}$ which extends the λ-terms and is given by the following grammars:

$$\mathcal{T} \;:=\; \mathcal{X} \mid \lambda\mathcal{X}.\mathcal{T} \mid (\mathcal{T}\ \ \mathcal{E}) \mid \langle \mathcal{T}, \mathcal{T}\rangle \mid \omega_1\mathcal{T} \mid \omega_2\mathcal{T} \mid \mu\mathcal{A}.\mathcal{T} \mid (\mathcal{A}\ \ \mathcal{T})$$
$$\mathcal{E} \;:=\; \mathcal{T} \mid \pi_1 \mid \ \pi_2 \mid [\mathcal{X}.\mathcal{T}, \mathcal{X}.\mathcal{T}]$$

An element of the set $\mathcal{E}$ is said to be an $\mathcal{E}$-term.

(ii) The meaning of the new constructors is given by the typing rules below where Γ (resp. Δ) is a context, i.e. a set of declarations of the form $x : A$ (resp. $a : A$) where x is a λ-variable (resp. a is a μ-variable) and A is a formula.

$$\frac{}{\Gamma, x : A \ \vdash x : A \ ; \ \Delta} ax$$

$$\frac{\Gamma, x : A \vdash t : B; \Delta}{\Gamma \vdash \lambda x.t : A \to B; \Delta} \to_i \qquad \frac{\Gamma \vdash u : A \to B; \Delta \quad \Gamma \vdash v : A; \Delta}{\Gamma \vdash (u \ \ v) : B; \Delta} \to_e$$

$$\frac{\Gamma \vdash u : A; \Delta \quad \Gamma \vdash v : B; \Delta}{\Gamma \vdash \langle u, v \rangle : A \wedge B; \Delta} \wedge_i$$

$$\frac{\Gamma \vdash t : A \wedge B; \Delta}{\Gamma \vdash (t \ \ \pi_1) : A; \Delta} \wedge_e^1 \quad \frac{\Gamma \vdash t : A \wedge B; \Delta}{\Gamma \vdash (t \ \ \pi_2) : B; \Delta} \wedge_e^2$$

$$\frac{\Gamma \vdash t : A; \Delta}{\Gamma \vdash \omega_1 t : A \vee B; \Delta} \vee_i^1 \quad \frac{\Gamma \vdash t : B; \Delta}{\Gamma \vdash \omega_2 t : A \vee B; \Delta} \vee_i^2$$

$$\frac{\Gamma \vdash t : A \vee B; \Delta \quad \Gamma, x : A \vdash u : C; \Delta \quad \Gamma, y : B \vdash v : C; \Delta}{\Gamma \vdash (t \ \ [x.u, y.v]) : C; \Delta} \vee_e$$

$$\frac{\Gamma \vdash t : A; \Delta, a : A}{\Gamma \vdash (a \ \ t) : \bot; \Delta, a : A} abs_i \quad \frac{\Gamma \vdash t : \bot; \Delta, a : A}{\Gamma \vdash \mu a.t : A; \Delta} abs_e$$

(iii) The cut-elimination procedure corresponds to the reduction rules given below. They are those we need to the subformula property.
- $(\lambda x.u \ \ v) \triangleright u[x := v]$
- $(\langle t_1, t_2 \rangle \ \ \pi_i) \triangleright t_i$
- $(\omega_i t \ \ [x_1.u_1, x_2.u_2]) \triangleright u_i[x_i := t]$
- $((t \ \ [x_1.u_1, x_2.u_2]) \ \ \varepsilon) \triangleright (t \ \ [x_1.(u_1 \ \varepsilon), x_2.(u_2 \ \varepsilon)])$
- $(\mu a.t \ \ \varepsilon) \triangleright \mu a.t[a :=^* \varepsilon]$.

  where $t[a :=^* \varepsilon]$ is obtained from t by replacing inductively each subterm in the form $(a \ v)$ by $(a \ (v \ \varepsilon))$.

(iv) Let t and t' be $\mathcal{E}$-terms. The notation $t \triangleright t'$ means that t reduces to t' by using one step of the reduction rules given above. Similarly, $t \triangleright^* t'$ means that t reduces to t' by using some steps of the reduction rules given above.

The following result is straightforward

Theorem 2.2 *(Subject reduction) If $\Gamma \vdash t : A; \Delta$ and $t \triangleright^* t'$, then $\Gamma \vdash t' : A; \Delta$.*

We have also the following properties (see [1], [2], [3], [8] and [9]).

Theorem 2.3 *(Confluence) If $t \triangleright^* t_1$ and $t \triangleright^* t_2$, then there exists t_3 such that $t_1 \triangleright^* t_3$ and $t_2 \triangleright^* t_3$.*

Theorem 2.4 *(Strong normalization) If $\Gamma \vdash t : A; \Delta$, then t is strongly normalizable.*

3 The semantics

Definition 3.1 (i) We denote by $\mathcal{E}^{<\omega}$ the set of finite sequences of $\mathcal{E}$-terms. The empty sequence is denoted by $\emptyset$.

(ii) We denote by $\bar{w}$ the sequence $w_1 w_2 ... w_n$. If $\bar{w} = w_1 w_2 ... w_n$, then $(t\ \bar{w})$ is t if $n = 0$ and $((t\ w_1)\ w_2 ... w_n)$ if $n \neq 0$. The term $t[a :=^* \bar{w}]$ is the term obtained from t by replacing inductively each subterm in the form $(a\ v)$ by $(a\ (v\ \bar{w}))$.

(iii) A set of terms S is said to be μ-saturated iff:
- For each terms u and v, if $u \in S$ and $v \triangleright^* u$, then $v \in S$.
- For each $a \in \mathcal{A}$ and for each $t \in S$, $\mu a.t \in S$ and $(a\ t) \in S$.

(iv) Consider two sets of terms K, L and a μ-saturated set S, we define new sets of terms:
- $K \to L = \{t\ /\ (t\ u) \in L$, for each $u \in K\}$.
- $K \wedge L = \{t\ /\ (t\ \pi_1) \in K$ and $(t\ \pi_2) \in L\}$.
- $K \vee L = \{t\ /$ for each u, v: if (for each $r \in K$,$s \in L$: $u[x := r] \in S$ and $v[y := s] \in S$), then $(t\ [x.u, y.v]) \in S\}$.

(v) Let S be a μ-saturated set and $\{R_i\}_{i \in I}$ subsets of terms such that $R_i = X_i \to S$ for certains $X_i \subseteq \mathcal{E}^{<\omega}$. A model $\mathcal{M} = \langle S; \{R_i\}_{i \in I}\rangle$ is the smallest set of subsets of terms containing S and R_i and closed under constructors $\to$, $\wedge$ and $\vee$.

Lemma 3.2 *Let $\mathcal{M} = \langle S; \{R_i\}_{i \in I}\rangle$ be a model and $G \in \mathcal{M}$.*
There exists a set $X \subseteq \mathcal{E}^{<\omega}$ such that $G = X \to S$.

Proof By induction on G.

- $G = S$: Take $X = \{\emptyset\}$, it is clear that $S = \{\emptyset\} \to S$.
- $G = G_1 \to G_2$: We have $G_2 = X_2 \to S$ for a certain set X_2. Take $X = \{u\ \bar{v}\ /\ u \in G_1, \bar{v} \in X_2\}$. We can easly check that $G = X \to S$.
- $G = G_1 \wedge G_2$: Similar to the previous case.
- $G = G_1 \vee G_2$: Take $X = \{[x.u, y.v]\ /$ for each $r \in G_1$ and $s \in G_2$, $u[x := r] \in S$ and $v[y := s] \in S\}$. By definition $G = X \to S$.

$\square$

Definition 3.3 Let $\mathcal{M} = \langle S; \{R_i\}_{i \in I}\rangle$ be a model and $G \in \mathcal{M}$, we define the set $G^\perp = \cup\{X\ /\ G = X \to S\}$.

Lemma 3.4 *Let $\mathcal{M} = \langle S; \{R_i\}_{i \in I}\rangle$ be a model and $G \in \mathcal{M}$.*
We have $G = G^\perp \to S$ ($G^\perp$ is the greatest X such that $G = X \to S$).

Proof This comes from the fact that: if, for every $j \in J$, $G = X_j \to S$, then $G = \cup_{j \in J} X_j \to S$. $\square$

Definition 3.5 (i) Let $\mathcal{M} = \langle S; \{R_i\}_{i \in I}\rangle$ be a model. An $\mathcal{M}$-interpretation I is an application from the set of propositional variables to $\mathcal{M}$ which we extend for any type as follows:

- $I(\perp) = S$
- $I(A \to B) = I(A) \to I(B)$.
- $I(A \wedge B) = I(A) \wedge I(B)$.
- $I(A \vee B) = I(A) \vee I(B)$.

The set $|A|_{\mathcal{M}} = \cap\{I(A) \ / \ I$ an $\mathcal{M}$-interpretation$\}$ is the interpretation of A in $\mathcal{M}$.

(ii) The set $|A| = \cap\{|A|_{\mathcal{M}} \ / \ \mathcal{M}$ a model$\}$ is the interpretation of A.

Lemma 3.6 *(Adequation lemma) Let $\mathcal{M} = \langle S; \{R_i\}_{i \in I}\rangle$ be a model, I a $\mathcal{M}$-interpretation, $\Gamma = \{x_i : A_i\}_{1 \leq i \leq n}$, $\Delta = \{a_j : B_j\}_{1 \leq j \leq m}$, $u_i \in I(A_i)$, $\bar{v_j} \in I(B_j)^{\perp}$.*

If $\Gamma \vdash t : A; \Delta$, then $t[x_1 := u_1, ..., x_n := u_n, a_1 :=^* \bar{v_1}, ..., a_m :=^* \bar{v_m}] \in I(A)$.

Proof Let us denote by s' the term

$s[x_1 := u_1, ..., x_n := u_n, a_1 :=^* \bar{v_1}, ..., a_m :=^* \bar{v_m}]$.

The proof is by induction on the derivation, we consider the last rule:

(i) ax, $\to_e$ and $\wedge_e$: Easy.

(ii) $\to_i$: In this case $t = \lambda x.u$ and $A = B \to C$ such that $\Gamma, x : B \vdash u : C \ ; \Delta$. By induction hypothesis, $u'[x := v] \in I(C) = I(C)^{\perp} \to S$ for each $v \in I(B)$, then $(u'[x := v] \ \bar{w}) \in S$ for each $\bar{w} \in I(C)^{\perp}$, hence $((\lambda x.u' \ v) \ \bar{w}) \in S$ because $((\lambda x.u' \ v) \ \bar{w}) \rhd^* (u'[x := v] \ \bar{w})$. Therefore $t' = \lambda x.u' \in I(B) \to I(C) = I(A)$.

(iii) $\wedge_i$ and $\vee_i^j$: A similar proof.

(iv) $\vee_e$: In this case $t = (t_1 \ [x.u, y.v])$ with $(\Gamma \vdash t_1 : B \vee C; \Delta)$, $(\Gamma, x : B \vdash u : A; \Delta)$ and $(\Gamma, y : C \vdash v : A; \Delta)$. Let $r \in I(B)$ and $s \in I(C)$, by induction hypothesis, $t_1' \in I(B) \vee I(C)$, $u'[x := r] \in I(A)$ and $v'[y := s] \in I(A)$. Let $\bar{w} \in I(A)^{\perp}$, then $(u'[x := r] \ \bar{w}) \in S$ and $(v'[y := s] \ \bar{w}) \in S$, hence $(t_1' \ [x.(u' \ \bar{w}), y.(v' \ \bar{w})]) \in S$, since $((t_1' \ [x.u', y.v')] \ \bar{w}) \rhd^* (t_1' \ [x.(u' \ \bar{w}), y.(v' \ \bar{w})])$ then $((t_1' \ [x.u', y.v')] \ \bar{w}) \in S$. Therefore $t' = (t_1' \ [x.u', y.v']) \in I(A)$.

(v) abs_e: In this case $t = \mu a.t_1$ and $\Gamma \vdash t_1 :\perp \ ; \Delta', a : A$. Let $\bar{v} \in I(A)^{\perp}$. It suffies to prove that $(\mu a.t_1' \ \bar{v}) \in S$. By induction hypothesis, $t_1'[a :=^* \bar{v}] \in I(\perp) = S$, then $\mu a.t_1'[a :=^* \bar{v}] \in S$ and $(\mu a.t_1' \ \bar{v}) \in S$.

(vi) abs_i: In this case $t = (a_j\ u)$ and $\Gamma \vdash u : B_j; \Delta', a_j : B_j$. We have to prove that $t' \in S$. By induction hypothesis $u' \in I(B_j)$, then $(u'\ \bar{v}_j) \in S$, hence $t' = (a\ (u'\ \bar{v}_j)) \in S$.

□

Theorem 3.7 *(Correctness theorem) If $\vdash t : A$, then $t \in |A|$.*

Proof Immediately from the previous lemma. □

4 The operational behaviors of some typed terms

The following results are some applications of the correctness theorem.

Definition 4.1 Let t be a term. We denote M_t the smallest set containing t such that: if $u \in M_t$ and $a \in \mathcal{A}$, then $\mu a.u \in M_t$ and $(a\ u) \in M_t$. Each element of M_t is denoted $\underline{\mu}.t$. For exemple, the term $\mu a.\mu b.(a\ (b\ (\mu c.(a\ \mu d.t))))$ is denoted by $\underline{\mu}.t$.

In the next of the paper, the letter P denotes a propositional variable which represents an arbitrary type.

4.1 Terms of type $\perp\rightarrow P$ "Ex falso sequitur quodlibet"

Example 4.2 Let $\mathcal{T} = \lambda z.\mu a.z$. We have $\mathcal{T} :\perp\rightarrow P$ and for every term t and $\bar{u} \in \mathcal{T}^{<\omega}$, $((\mathcal{T}\ t)\ \bar{u}) \rhd^* \mu a.t$.

Remark 4.3 The term $(\mathcal{T}\ t)$ modelizes an instruction like `exit`(t) (`exit` is to be understood as in the C programming language). In the reduction of a term, if the sub-term $(\mathcal{T}\ t)$ appears in head position (the term has the form $((\mathcal{T}\ t)\ \bar{u})$), then after some reductions, we obtain t as result.

The general operational behavior of terms of type $\perp\rightarrow P$ is given in the following theorem:

Theorem 4.4 *Let T be a closed term of type $\perp\rightarrow P$, then for every term t and $\bar{u} \in \mathcal{E}^{<\omega}$, $((T\ t)\ \bar{u}) \rhd^* \underline{\mu}.t$.*

Proof Let t be a term and $\bar{u} \in \mathcal{E}^{<\omega}$. Take $S = \{v\ /\ v \rhd^* \underline{\mu}.t\}$ and $R = \{\bar{u}\} \to S$. It is clear that S is μ-saturated set and $t \in S$. Let $\mathcal{M} = \langle S; R\rangle$ and I an $\mathcal{M}$-interpretation such that $I(P) = R$. By the theorem 3.7, we have $T \in S \to (\{\bar{u}\} \to S)$, then $((T\ t)\ \bar{u}) \in S$ and $((T\ t)\ \bar{u}) \rhd^* \underline{\mu}.t$. □

4.2 Terms of type $(\neg P \to P) \to P$ "Pierce law"

Example 4.5 Let $\mathcal{C}_1 = \lambda z.\mu a.(a\,(z\ \lambda y.(a\,y)))$ and
$\mathcal{C}_2 = \lambda z.\mu a.(a\,(z\ (\lambda x.a(z\ \lambda y.(a\,x)))))$.
We have $\vdash \mathcal{C}_i : (\neg P \to P) \to P$ for $i \in \{1,2\}$.
Let u, v_1, v_2 be terms and $\bar{t} \in \mathcal{E}^{<\omega}$, we have :
$((\mathcal{C}_1\ u)\ \bar{t}) \triangleright^* \mu a.a\ ((u\,\theta_1)\,\bar{t})$ and $(\theta_1\,v_1) \triangleright^* (a\,(v_1\ \bar{t}))$
and
$((\mathcal{C}_2\ u)\ \bar{t}) \triangleright^* \mu a.((a\,((u\,\theta_1)\,\bar{t}))\,\bar{t})$, $(\theta_1\ v_1) \triangleright^* (a\,((u\,\theta_2)\,\bar{t}))$ and $(\theta_2\ v_2) \triangleright^* (a\,(v_1\,\bar{t}))$.

Remark 4.6 The term $\mathcal{C}_1$ allows to modelizing the `Call/cc` instruction in the Scheme functional programming language.

The following theorem describes the general operational behavior of terms with type $(\neg P \to P) \to P$.

Theorem 4.7 *Let T be a closed term of type $(\neg P \to P) \to P$, then for every term u and $\bar{t} \in \mathcal{E}^{<\omega}$, there exist $m \in \mathbb{N}$ and terms $\theta_1, ..., \theta_m$ such that for every terms $v_1, ..., v_m$, we have:*
$((T\ u)\ \bar{t}) \triangleright^* \underline{\mu}.((u\,\theta_1)\,\bar{t})$
$(\theta_i\ v_i) \triangleright^* \underline{\mu}.((u\,\theta_{i+1})\,\bar{t})$ *for every* $1 \le i \le m-1$
$(\theta_m\ v_m) \triangleright^* \underline{\mu}.(v_{i_0}\,\bar{t})$ *for a certain* $1 \le i_0 \le m$

Proof Let u be a λ-variable and $\bar{t} \in \mathcal{E}^{<\omega}$. Take $S = \{t \ /\ \exists m \ge 0, \exists \theta_1, ..., \theta_m : t \triangleright^* \underline{\mu}.((u\ \theta_1)\,\bar{t}), (\theta_i\ v_i) \triangleright^* \underline{\mu}.((u\,\theta_{i+1})\,\bar{t})$ for every $1 \le i \le m-1$ and $(\theta_m\ v_m) \triangleright^* \underline{\mu}.(v_{i_0}\bar{t})$ for a certain $1 \le i_0 \le m\}$ and $R = \{\bar{t}\} \to S$. It is clear that S is a μ-saturated set. Let $\mathcal{M} = \langle S; R \rangle$ and an $\mathcal{M}$-interpretation I such that $I(P) = R$. By the theorem 3.7, $T \in [(R \to S) \to R] \to (\{\bar{t}\} \to S)$. It is suffies to check that $u \in (R \to S) \to R$. For this, we take $\theta \in (R \to S)$ and we prove that $(u\ \theta) \in R$ i.e. $((u\ \theta)\ \bar{t}) \in S$. But by the definition of S, it suffies to have $(\theta\ v_i) \in S$, which is true since the terms $v_i \in R$, because $(v_i\ \bar{t}) \in S$. $\square$

4.3 Terms of type $\neg P \vee P$ "Tertium non datur"

Example 4.8 Let $\mathcal{W} = \mu b.(b\,\omega_1\mu a.(b\,\omega_2\lambda y.(a\,y)))$. We have $\vdash \mathcal{W} : \neg P \vee P$.
Let x_1, x_2 be λ-variables, u_1, u_2, v terms and $\bar{t} \in \mathcal{E}^{<\omega}$. We have:
$(\mathcal{W}\,[x_1.u_1, x_2.u_2]) \triangleright^* \mu b.(b\ u_1\,[x_1 := \theta_1^1])$
$(\theta_1^1\,\bar{t}) \triangleright^* \mu a.(b\,u_2\,[x_2 := \theta_2^2])$
$(\theta_2^2\,v) \triangleright^* (a(v\,\bar{t}))$
where $\theta_1^1 = \mu a.(b\ (\omega_2\lambda y.(a\ y)\ [x_1.u_1, x_2.u_2]))$ and $\theta_2^2 = \lambda y.(a\ (y\ \bar{t}))$.

Remark 4.9 The term $\mathcal{W}$ allows to modelizing the `try`...`with`... instruction in the Caml programming language.

The following theorem gives the behavior of all terms with type $\neg P \vee P$.

Theorem 4.10 *Let T be a closed term of type $\neg P \vee P$, then for every λ-variables x_1, x_2 and terms u_1, u_2 and $(\bar{t_n})_{n\geq 1}$ a sequence of $\mathcal{E}^{<\omega}$, there exist $m \in \mathbb{N}$ and terms $\theta^i_1, ..., \theta^i_m$ $1 \leq i \leq 2$ such that for all terms $v_1, ..., v_m$, we have:*

$(T\,[x_1.u_1, x_2.u_2]) \triangleright^* \underline{\mu}.u_i[x_i := \theta^i_1]$

$(\theta^1_j\,\bar{t_j}) \triangleright^* \underline{\mu}.u_i[x_i := \theta^i_{j+1}]$ *for all* $1 \leq j \leq m-1$

$(\theta^2_j\,v_j) \triangleright^* \underline{\mu}.u_i[x_i := \theta^i_{j+1}]$ *for all* $1 \leq j \leq m-1$

$(\theta^1_m \bar{t_m}) \triangleright^* \underline{\mu}.(v_p\,\bar{t_q})$ *for a certain* $1 \leq p \leq m$ *and a certain* $1 \leq q \leq m$

$(\theta^2_m\,v_m) \triangleright^* \underline{\mu}.(v_p\,\bar{t_q})$ *for a certain* $1 \leq p \leq m$ *and a certain* $1 \leq q \leq m$

Proof Let u_1, u_2 be terms and $(\bar{t_n})_{n\geq 1}$ a sequence of $\mathcal{E}^{<\omega}$. Take then $S = \{t \,/\, \exists m \geq 0, \exists \theta^i_1, ..., \theta^i_m\ 1 \leq i \leq 2 : t \triangleright^* \underline{\mu}.u_i[x_i := \theta^i_1]$, $(\theta^1_j\,\bar{t_j}) \triangleright^* \underline{\mu}.u_i[x_i := \theta^i_{j+1}]$ for all $1 \leq j \leq m-1$, $(\theta^2_j\,v_j) \triangleright^* \underline{\mu}.u_i[x_i := \theta^i_{j+1}]$ for all $1 \leq j \leq m-1$, $(\theta^1_m\,\bar{t_m}) \triangleright^* \underline{\mu}.v_p(\bar{t_q})$ for certain $(1 \leq p \leq m$ and $1 \leq q \leq m)$ and $(\theta^2_m\,v_m) \triangleright^* \underline{\mu}.(v_p\,\bar{t_q})$ for certain $(1 \leq p \leq m$ and $1 \leq q \leq m)\}$. $R = \{\bar{t_1}, ..., \bar{t_n}\} \to S$. By definition S is a μ-saturated set. Let $\mathcal{M} = \langle S; R\rangle$ and an $\mathcal{M}$-interpretation I such that $I(P) = R$. By the theorem 3.7, $T \in [R \to S] \vee R$. Let $\theta \in R$, then, for all i, $(\theta\ \bar{t_i}) \in S$. Let $\theta' \in R \to S$, hence $(\theta'\ v_i) \in S$ since $v_i \in R$ (because $(v_i\ \bar{t_i}) \in S$), therefore $(T\,[x_1.u_1, x_2.u_2]) \in S$. □

References

[1] Y. Andou. *Church-Rosser property of simple reduction for full first-order classical natural deduction.* Annals of Pure and Applied Logic 119 (2003) 225-237.

[2] R. David and K. Nour. *A short proof of the Strong Normalization of Classical Natural Deduction with Disjunction.* Journal of Symbolic Logic, vol 68, num 4, pp 1277-1288, 2003.

[3] P. de Groote. *Strong normalization of classical natural deduction with disjunction.* In 5th International Conference on Typed Lambda Calculi and Applications, TLCA'01. LNCS (2044), pp. 182-196. Springer Verlag, 2001.

[4] G. Gentzen. *Recherches sur la déduction logique.* Press Universitaires de France, 1955. Traduction et commentaires par R. Feys et J. Ladrière.

[5] J.-L. Krivine. *Lambda calcul, types et modèle.* Masson, Paris, 1990.

[6] R. Matthes. *Inductive Constructions for Classical Natural Deduction.* To be submitted.

[7] K. Nour. *Mixed Logic and Storage Operators.* Archive for Mathematical Logic, vol 39, pp. 261-280, 2000.

[8] K. Nour and K. Saber. *A semantical proof of strong normalization natural classical deduction with disjunction.* Manuscript, 2004.

[9] K. Nour and K. Saber. *Church- Rosser property of full propositional classical natural deduction.* Manuscrit, 2004.

[10] M. Parigot. *$\lambda\mu$-calculus: an algorithmic interpretation of classical natural deduction.* Lecture Notes in Artificial Intelligence, 624, Springer Verlag, 1992.

[11] M. Parigot. *Proofs of strong normalization for second order classical natural deduction.* Journal of Symbolic Logic, 62 (4), pp. 1461-1479, 1997.