Complete reducibility candidates

Denis Cousineau

Complete reducibility candidates

Denis Cousineau
http://www.denis.cousineau.eu

TypiCal - INRIA Saclay Île de France - Ecole Polytechnique

Abstract. Deduction modulo is an extension of first-order predicate logic where axioms are replaced by a congruence relation on propositions and where many theories, such as arithmetic, simple type theory and some variants of set theory, can be expressed. An important question in deduction modulo is to find a condition of the theories that have the strong normalization property. Dowek and Werner have given a semantic sufficient condition for a theory to have the strong normalization property: they have proved a "soundness" theorem of the form: if a theory has a model (of a particular form) then it has the strong normalization property. In this paper, we refine their notion of model in a way allowing not only to prove soundness, but also completeness: if a theory has the strong normalization property, then it has a model of this form. The key idea of our model construction is a refinement of Girard's notion of reducibility candidates. By providing a sound and complete semantics for theories having the strong normalization property, this paper contributes to explore the idea that strong normalization is not only a proof-theoretic notion, but also a model-theoretic one.

In this paper, we define a sound and complete semantics for theories having the strong normalization property in minimal deduction modulo.

Deduction modulo [5] is a logical framework, based on Natural Deduction where axioms are replaced by a congruence relation on propositions, allowing to express proofs of many theories like arithmetic [9], simple type theory [6], some variants of set theory [7], etc... The absence of axioms ensures the fact that all cut-free proofs end with an introduction rule, as in usual Natural Deduction. Hence the cut elimination property of a theory entails its consistency. In this framework, cuts in proofs are represented by β-redexes, and the elimination of a cut, by β-reduction. Therefore strong normalization of the β-reduction ensures the cut elimination property of the corresponding theory, and furthermore its consistency.

The usual tool to prove strong normalization is called reducibility candidates. The notion of reducibility candidates was first introduced by J.Y. Girard [11], following the work of W.W. Tait [18]. We can see *a posteriori* their work as proofs of strong normalization, obtained by the existence of a $\mathcal{C}$-valued model, where $\mathcal{C}$ is the algebra of reducibility candidates. This work has been extended to, at least, two non-trivial logical frameworks: Pure Type Systems by P.A. Melliès and B. Werner [16], and Deduction modulo by G. Dowek and B. Werner [8]. By non-trivial, we mean that these logical frameworks can express strongly normalizing

and not strongly normalizing theories. In other words, they proved that having a $\mathcal{C}$-valued model is a sufficient semantic condition of strongly normalizing theories expressed in Pure Type Systems and Deduction modulo.

Reducibility candidates are a very useful tool to prove strong normalization but we may wonder if it is an exhaustive test. Is having a $\mathcal{C}$-valued model also a necessary semantic condition for the strong normalization property, *i.e.* do all strongly normalizing theories expressed in Deduction modulo or Pure Type Systems have a $\mathcal{C}$-valued model ?

In this paper, we exhibit a new algebra $\mathcal{C}'$ which is a refinement of $\mathcal{C}$ and we prove that having a $\mathcal{C}'$-valued model is still a sufficient semantic condition of strongly normalizing theories expressed in minimal deduction modulo. And moreover, that it is also a necessary semantic condition of strongly normalizing theories.

1 Minimal deduction modulo and truth values algebras

Let us first define the logical framework we use in this paper: minimal deduction modulo is deduction modulo with only two connectives : $\Rightarrow$ and $\forall$.

1.1 Minimal deduction modulo

Syntax Our language of propositions is that of multi-sorted first-order logic. Propositions are built up from predicates given by a many-sorted predicate language $\langle \mathbb{T}, \mathbb{F}, \mathbb{P} \rangle$ which contains a set $\mathbb{T}$ of *sorts*, a set $\mathbb{F}$ of function symbols, and a set $\mathbb{P}$ of predicate symbols. Given a infinite set of variables for each sort T in $\mathbb{T}$, we build terms and atomic propositions, by using the following rules: variables of sort T are terms of sort T; if f is a function symbol of rank $\langle T_1, \ldots, T_n, U \rangle$ and $t_1, \ldots, t_n$ are respectively terms of sort $T_1, \ldots, T_n$, then $f\ t_1 \ldots t_n$ is a term of sort U; if P is a predicate symbol of rank $\langle T_1, \ldots, T_n \rangle$ and $t_1, \ldots, t_n$ are respectively terms of sort $T_1, \ldots, T_n$, then $P\ t_1 \ldots t_n$ is an *atomic proposition*.

Propositions are built-up from atomic propositions with the usual connective $\Rightarrow$ and quantifier $\forall$: an atomic proposition is a proposition and if A and B are propositions and x is a term-variable then $\forall x.A$ and $A \Rightarrow B$ are propositions. Remark that, implicitly, quantification in $\forall x.A$ is restricted over the sort of the variable x. We call $\mathcal{P}$ the set of propositions.

In this logical framework, we provide proof-terms which represent constructors of proof derivations. Each proof-term construction corresponds to a natural deduction rule: proof-terms of the form α express proofs built with the axiom rule, proof-terms of the form $\lambda\alpha.\pi$ and $(\pi\ \pi')$ express proofs built respectively with the introduction and elimination rules of the implication and proof-terms of the form $\lambda x.\pi$ and $(\pi\ t)$ express proofs built with the introduction and elimination rules of the universal quantifier.

We call *neutral* those proof-terms that are not abstractions *i.e.* the proof-terms of the form α, $(\pi\pi')$ or (πt).

Notice that in this language, proof-terms can contain both term variables (written $x, y, \ldots$) and proof variables (written $\alpha, \beta, \ldots$). Terms are written $t, u, \ldots$ while proof-terms are written $\pi, \rho, \ldots$

Typing rules We call *contexts*, lists of declarations $[\alpha : A]$ where α is a proof-variable and A is a proposition, such that each variable is declared at most once. In this way, we only consider *well formed* contexts, therefore we have to rename variables when concatening contexts: the only proof-variables that two concatened contexts can share have to be declared proofs of the same proposition. Notice that as we declare only proof-variables and not term-variables in contexts, the concatenation of two contexts is, modulo renaming, always a context.

Given a congruence relation on propositions $\equiv$, we define typing rules as usual, in deduction modulo:

$$\frac{}{\Gamma, \alpha : A \vdash_{\equiv} \alpha : B} \quad A \equiv B \quad \text{(axiom)}$$

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

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

$$\frac{\Gamma \vdash_{\equiv} \pi : A}{\Gamma \vdash_{\equiv} \lambda x.\pi : B} \quad B \equiv \forall x.A, \quad x \notin FV(\Gamma) \quad (\forall\text{-intro})$$

$$\frac{\Gamma \vdash_{\equiv} \pi : B}{\Gamma \vdash_{\equiv} \pi\ t : C} \quad B \equiv \forall x.A, \quad C \equiv (t/x)A, \quad t \text{ has the sort of } x \quad (\forall\text{-elim})$$

Fig. 1. Typing rules

The point is that a proposition can be replaced by an equivalent one, at any place in a proof derivation.

Proof reduction rules and strong normalization In deduction modulo, the process of cut elimination is modeled by β-reduction. We consider the contextual closure of the reduction rules given figure 2. These rules correspond to proof reduction in natural deduction.

$$(\lambda\alpha.\pi\ \pi') \to (\pi'/\alpha)\pi$$
$$(\lambda x.\pi\ t) \to (t/x)\pi$$

Fig. 2. Proof reduction rules

We write $(\pi'/\alpha)\pi$ (resp. $(t/x)\pi$) the substitution of α (resp. x) by π' (resp. t) in π.
A proof-term is said to be *normal* if it contains no redex and *strongly normalizing* if all reduction sequences starting from this proof-term are finite. We write SN for the set of strongly normalizing proof-terms.
A proof-term is called *isolated* if it is neutral and only reduces to neutral proof-terms (*i.e* it never reduces to an abstraction, in any number of reduction steps).

Theories expressed in minimal deduction modulo A theory expressed in minimal deduction modulo is defined by a many-sorted language in predicate logic $\langle \mathbb{T}, \mathbb{F}, \mathbb{P} \rangle$ and a congruence relation $\equiv$ on propositions of the associated many-sorted logic. Given a theory $\langle \mathbb{T}, \mathbb{F}, \mathbb{P} \rangle_{\equiv}$, we will write $\vdash$ for $\vdash_{\equiv}$.

1.2 Language dependent truth values algebras

Truth values algebras (TVAs) are an extension of Heyting algebras, defined by G. Dowek in [4], which provide an algebraic setting to study consistency and cut elimination of theories expressed in deduction modulo. We use in this paper *language-dependent truth values algebras* (LDTVAs) which are both a simplification and a refinement of TVAs. They are first a simplification because they do not include the notion of *positive* truth values, as we only consider theories where axioms are expressed within the congruence relation and not given by inference rules. They also are a refinement in the sense that we use a functional interpretation of the connective $\forall$ (as in [10]), instead of a set-theoritic intersection. See [2] for details.
For all sorts T, we write $\hat{T}$, the set of closed terms (terms which do not contain variables) of sort T.

LDTVAS Given a many-sorted language in predicate logic $\langle \mathbb{T}, \mathbb{F}, \mathbb{P} \rangle$, a LDTVA for this language is an algebraic structure $\langle \mathcal{B}, \hat{\Rightarrow}, (\hat{\mathcal{A}}_T), (\hat{\forall}_T) \rangle$ where $\mathcal{B}$ is a set (called the *domain*), $\hat{\Rightarrow}$ is a function from $\mathcal{B} \times \mathcal{B}$ to $\mathcal{B}$, and for all sorts T of $\mathbb{T}$, $\hat{\mathcal{A}}_T$ is a set of functions from $\hat{T}$ to $\mathcal{B}$ and $\hat{\forall}_T$ is a function from $\hat{\mathcal{A}}_T$ to $\mathcal{B}$.
Notice that we will write $\mathcal{B}$ both for denominating the LDTVA $\mathcal{B}$ and its domain.

A *valuation* φ is a substitution mapping term-variables of a sort T to closed terms of sort T. For all propositions A, we call $\text{VAL}(A)$ the set of valuations whose domain contains the set of free variables of A. And we write $\text{DOM}(\varphi)$ the domain of a valuation φ. For all $A \in \mathcal{P}$ and $\varphi \in \text{VAL}(A)$, we write A_φ the result of the valuation φ on A.

Models We call *$\mathcal{B}$-valued interpretations* those functions which map every ordered pair of a proposition A and a valuation in $\text{VAL}(A)$ to an element of the domain of the LDTVA $\mathcal{B}$.

A $\mathcal{B}$-valued interpretation $[\![.]\!]_.$ is a *$\mathcal{B}$-valued model* if and only if:
for all $A, B \in \mathcal{P}$, $\varphi \in \text{VAL}(A \Rightarrow B)$, x of sort T, $t \in \hat{T}$ and $\psi \in \text{VAL}(\forall x.A)$,
$[\![A \Rightarrow B]\!]_\varphi = [\![A]\!]_\varphi \hat{\Rightarrow} [\![B]\!]_\varphi, \quad (t \mapsto [\![A]\!]_{\psi + \langle x,t \rangle}) \in \hat{\mathcal{A}}_T,$
$[\![\forall x.A]\!]_\psi = \hat{\forall}_T(t \mapsto [\![A]\!]_{\psi+\langle x,t\rangle}),$ and $[\![(t/x)A]\!]_\psi = [\![A]\!]_{\psi+\langle x,t\rangle}.$

We also say that the $\mathcal{B}$-valued interpretation $[\![.]\!]_.$ is *adapted to the connectives*.

A $\mathcal{B}$-valued model $[\![.]\!]_.$ is a *model of the theory* $\langle \mathbb{T}, \mathbb{F}, \mathbb{P} \rangle_{\equiv}$ if and only if:
for all $A, A' \in \mathcal{P}$, $\varphi \in \textsc{Val}(A)$ and $\psi \in \textsc{Val}(A')$, if $A_\varphi \equiv A'_\psi$, then $[\![A]\!]_\varphi = [\![A']\!]_\psi$
We also say that the $\mathcal{B}$-valued model $[\![.]\!]_.$ is *adapted to the congruence*.

Morphisms of LDTVAS are defined as usual and morphisms from $\mathcal{B}_1$ to $\mathcal{B}_2$ induce a mapping from $\mathcal{B}_1$-valued models to $\mathcal{B}_2$-valued models of a same theory.

2 Reducibility candidates

2.1 The adequation lemma

The reducibility candidates were first defined by J.Y. Girard in its proof of strong normalization of system F, following the work of W.W. Tait for system T. The main idea of this kind of proofs of strong normalization is to define a class of sets of proof-terms which contains all proofs of propositions and contains only strongly normalizing proof-terms. More precisely, the point is to first assign to each proposition A, a set of strongly normalizing proof-terms $\mathcal{R}_A$. And then prove an *adequation lemma*: every proof of A is in $\mathcal{R}_A$ (hence is strongly normalizing). In order to prove this adequation lemma, it is sufficient to prove the following statements: each $\mathcal{R}_A$ is stable by β-reduction; if a neutral proof reduces only to elements of a $\mathcal{R}_A$ then it is also in $\mathcal{R}_A$; and $\mathcal{R}_{A \Rightarrow B}$ is exactly the set of proof-terms which map proof-terms in $\mathcal{R}_A$ to proof-terms in $\mathcal{R}_B$ (and an analog property for each connective).

2.2 $\mathcal{C}$, the TVA of reducibility candidates

G. Dowek and B. Werner extended this kind of proofs of strong normalization to deduction modulo [8], by adding one more condition on reducibility candidates: the fact that if A and B are equivalent propositions, then $\mathcal{R}_A = \mathcal{R}_B$.

In [4], G. Dowek showed that reducibility candidates form a TVA $\mathcal{C}$ and reexpressed the previous proof in an algebraic view: having a $\mathcal{C}$-valued-model is a sufficient condition for a theory to be strongly normalizing. Like LDTVAs, TVAs are based on a domain and a function space $\Rightarrow$. In the case of $\mathcal{C}$, the domain is the set of sets E of proof-terms wich satisfy the usual properties of reducibility candidates:

(CR_1) $E \subseteq SN$
(CR_2) if $\pi \in E$ and $\pi \to \pi'$, then $\pi' \in E$
(CR_3) if π is neutral and all its (one-step) reducts are in E, then π is also in E

and $E \Rightarrow F$ is the set $\{\pi$, for all $\pi' \in E$, $\pi\pi' \in F\}$, for all $E, F \in \mathcal{C}$.

2.3 Not known to be complete

Reducibility candidates are sound, in the sense that having a $\mathcal{C}$-valued model is a sufficient condition for a theory to be strongly normalizing. But they are not known to be complete. In order to prove that having a $\mathcal{C}$-calued model is also a necessary condition for a theory to be strongly normalizing, the naive idea to prove the converse of the adequation lemma does not work: usual reducibility candidates are not *adapted to typing* as the reducibility candidate associated to a propostion A does not contain only proofs of A. In fact, because of the (CR_3) property, all reducibility candidates contain all neutral normal proof-terms (as they have no reduct). Hence $\alpha\alpha$ is in all reducibility candidates, whereas it cannot be the proof of a proposition in a strongly normalizing theory: if we can type $\alpha\alpha$ then there exists two propositions A and B and a context Γ such that $\Gamma \vdash \alpha : A$ and $\Gamma \vdash \alpha : A \Rightarrow B$. Therefore $A \equiv A \Rightarrow B$. Hence, if we write $\delta = \lambda\alpha.\alpha\alpha$, we have $\Gamma \vdash \delta : A \Rightarrow B$ and also $\Gamma \vdash \delta : A$, by equivalence of A and $A \Rightarrow B$. Finally we obtain $\Gamma \vdash \delta\delta : B$ whereas $\delta\delta$ is not normalizing as it reduces to itself in one step of β-reduction. This leads to define the notion of well-typed reducibility candidates, to avoid neutral normal unwell-typed proof-terms to be in all reducibility candidates.

3 Well-typed reducibility candidates

3.1 $\mathcal{C}_{\equiv}$, the tva of $\equiv$-well-typed reducibility candidates

In order to build reducibility candidates which contain only proofs of the associated proposition (i.e. adapted to typing), the first idea is to consider ordered pairs of a context and a proof-term, rather than only proof-terms (as a typing judgement associates such an ordered pair to a proposition). And then, restrict the (CR_3) property to well-typed terms: each reducibility candidates has an associated proposition and only proofs of this proposition can be added using the (CR_3) property. As typing depends on the congruence relation, the LDTVA $\mathcal{C}_{\equiv}$ of well-typed reducibility candidates depends on the theory we study. Given a congruence relation $\equiv$, we define $\mathcal{C}_{\equiv} = \langle \mathcal{C}_{\equiv}, \mathring{\Rightarrow}, (\mathring{\mathcal{A}}_T), (\mathring{\forall}_T) \rangle$ as: the domain $\mathcal{C}_{\equiv}$ is the set of sets E of ordered pairs of a context and a proof, wich satisfy the following properties:

($\mathrm{CR}_{\equiv}$) There exists A_E such that $\forall (\Gamma, \pi) \in E,\ \Gamma \vdash \pi : A_E$
($\mathrm{CR}_{1_{\equiv}}$) If $(\Gamma, \pi) \in E$, then $\pi \in SN$
($\mathrm{CR}_{2_{\equiv}}$) If $(\Gamma, \pi) \in E$, and $\pi \to \pi'$, then $(\Gamma, \pi') \in E$
($\mathrm{CR}_{3_{\equiv}}$) If π is neutral, $\Gamma \vdash \pi : A_E$,
and for all one-step reducts τ of π, $(\Gamma, \tau) \in E$, then $(\Gamma, \pi) \in E$.

$E \mathring{\Rightarrow} F = \{(\Gamma, \pi)$, such that for all $(\Gamma', \pi') \in E$, $(\Gamma\Gamma', \pi\pi') \in F\}$,
Remind that we may have to rename proof-variables in π' and Γ'.
$\mathring{\mathcal{A}}_T = \{f : \hat{T} \mapsto \mathcal{C}_{\equiv}$, such that there exists $A_f \in \mathcal{P}$ and x_f such that
for all $t \in \hat{T}$ and $(\Gamma, \pi) \in f(t),\ \Gamma \vdash \pi : (t/x_f)A_f\}$
and $\mathring{\forall}_T f = \{(\Gamma, \pi)$ such that for all $t \in \hat{T}$, $(\Gamma, \pi t) \in f(t)\}$.

The proof that this defines a LDTVA can be found in [2].

3.2 $\mathcal{C}_{\equiv}$-models are complete

Given this notion of reducibility candidates, we are able to define a $\mathcal{C}_{\equiv}$-valued interpretation which is a model when the associated theory is strongly normalizing.

A connectives-adapted interpretation We define a first $\mathcal{C}_{\equiv}$-interpretation $[.]_.$ built-up from the interpretation of atomic propositions, by $\overset{\circ}{\Rightarrow}$ and $\overset{\circ}{\forall}$.

$[P\ t_1 \dots t_n]_\varphi = \{(\Gamma, \pi)$ such that $\pi \in SN$ and $\Gamma \vdash \pi : (P\ t_1 \dots t_n)_\varphi\}$
$[B \Rightarrow C]_\varphi = [B]_\varphi \overset{\circ}{\Rightarrow} [C]_\varphi$
$[\forall x.B]_\varphi = \overset{\circ}{\forall}_T\ (t \mapsto [B]_{\varphi + \langle x,t\rangle})$

First we can notice that for all $A \in \mathcal{P}$ and $\varphi \in \text{VAL}(A)$, $[A]_\varphi$ is not empty as it contains $(\alpha : A_\varphi, \alpha)$. In fact, we can even prove that if π is isolated, strongly normalizing and $\Gamma \vdash \pi : A_\varphi$ then $(\Gamma, \pi) \in [A]_\varphi$.

This interpretation is, by construction, always adapted to typing (thanks to the definitions of $\overset{\circ}{\Rightarrow}$, $\overset{\circ}{\forall}$ and the atomic case) and adapted to the connectives, but it is not necessarily adapted to the congruence. In other words, $[.]_.$ is a model but not necessarily a model of the thory. Indeed in a theory which contains two atomic propositions P and Q such that $P \equiv (Q \Rightarrow Q)$ (notice that such a theory can be strongly normalizing), then for all valuations $\varphi \in \text{VAL}(P) \cap \text{VAL}(Q)$, we cannot prove $[P]_\varphi = [Q]_\varphi \overset{\circ}{\Rightarrow} [Q]_\varphi$. We have then to modify this interpretation to make it a $\mathcal{C}_{\equiv}$-valued model of $\langle \mathbb{T}, \mathbb{F}, \mathbb{P} \rangle_{\equiv}$ (i.e. also adapted to the congruence).

Adapting to the congruence Making this interpretation adapted to the congruence is not difficult: we just take the intersection of all the first interpretations of equivalent propositions. We then define a second interpretation $\lfloor . \rfloor_.$ as follows:

$\lfloor A \rfloor_\varphi = \bigcap_{A_\varphi \equiv A'_\psi} [A']_\psi$

Of course this interpretation is still adapted to typing and non-empty, but it is not necessarily adapted to the connectives anymore. The main point of this first completeness theorem is to prove that it is still adapted to the connectives when the theory is strongly normalizing. We proceed by contraposition, showing that if $\lfloor . \rfloor_.$ is not adapted to the connectives, then we can exhibit a well-typed proof-term which is not strongly normalizing. If there exists $A, B \in \mathcal{P}$, $\varphi \in \text{VAL}(A \Rightarrow B)$ or $\varphi' \in \text{VAL}(\forall x.A)$ with x of sort T, $x \notin \text{DOM}(\varphi')$ such that $\lfloor A \Rightarrow B \rfloor_\varphi \neq \lfloor A \rfloor_\varphi \overset{\circ}{\Rightarrow} \lfloor B \rfloor_\varphi$ or $\lfloor \forall x.A \rfloor_{\varphi'} \neq \overset{\circ}{\forall}_T (t \mapsto \lfloor A \rfloor_{\varphi' + \langle x,t \rangle})$, then there exists $C \in \mathcal{P}$, $\psi \in \text{VAL}(C)$ and π such that $\Gamma \vdash \pi : C_\psi$ and $(\Gamma, \pi) \notin \lfloor C \rfloor_\psi$. And we can then go backward to the first interpretation and exhibit $D \in \mathcal{P}$, $\psi \in \text{VAL}(D)$ and π' such that $\Gamma \vdash \pi' : D_\psi$ and $(\Gamma, \pi') \notin [D]_\psi$. Finally, we can prove that if there exists such a D, then there also exists an atomic propostion which satisfies the same property. And we conclude by definition of the first interpretation on atomic propositions, that there exists a well-typed proof-term which is not strongly normalizing, i.e. the theory is not strongly normalizing.

3.3 $\mathcal{C}_{\equiv}$-models are also sound but depend on the theory

We proved that having a $\mathcal{C}_{\equiv}$-valued model is a complete condition for a theory to be strongly normalizing. It is not difficult to prove that it is also a sound

condition. Indeed, you can notice that in the adequation lemma, the fact that the considered proof-term is well-typed is assumed. We can therefore use the ($\mathrm{CR}_{3_{\equiv}}$) property instead of the usual (CR_3), when adapting the proof of [4]. We get, this way, a first sound and complete condition for strongly normalizing theories. But this condition does not give a characterization of strong normalization as theories having some sort of model: as we said above, the algebra $\mathcal{C}_{\equiv}$ depends on the studied theory, and we have therefore defined a different criterium for each theory.

4 Theory-independent complete reducibility candidates

In this section we define a new LDTVA $\mathcal{C}'$ which does not depend on a theory anymore and we prove completeness of $\mathcal{C}'$-models by defining a morphism from each $\mathcal{C}_{\equiv}$ to $\mathcal{C}'$. In this section, we write $[\pi'/\alpha]\pi$ the substitution with capture of α by π' in π.

4.1 Another (CR_3) property

First, we naturally consider in this section sets of proof-terms again, and we do not mention contexts in the LDTVA we build, as we do not want to depend on typing anymore.
We define $\mathcal{C}' = \langle \mathcal{C}', \tilde{\Rightarrow}, (\tilde{\mathcal{A}}_T), (\tilde{\forall}_T) \rangle$ as follows: the domain of $\mathcal{C}'$ as sets E of proof-terms which satisfy the usual (CR_1), (CR_2) and another modified version of (CR_3):

(CR'_3) $E \neq \emptyset$ and for all $n \in \mathbb{N}$, for all $\nu, \mu_1, \ldots, \mu_n \in \mathcal{T}$, if
- for all $i \leq n$, μ_i is neutral and not normal,
- $\forall \rho_1, \ldots, \rho_n$ such that for all $i \leq n$, $\mu_i \to \rho_i$, $[\rho_i/\alpha_i]_{i \leq n}\nu \in E$

then $[\mu_i/\alpha_i]_{i \leq n}\nu \in E$.

And we use the usual interpretations of connectives:

$E \tilde{\Rightarrow} F = \{\pi \in SN \text{ such that for all } \pi' \in E,\ \pi\pi' \in F\}$

$\tilde{\mathcal{A}}_T = \hat{T} \mapsto \mathcal{C}' \quad \text{and} \quad \tilde{\forall}_T.f = \{\pi \text{ such that for all } t \in \hat{T},\ \pi t \in f(t)\}$

The not so simple proof that this defines a LDTVA can be found in [2].

This (CR'_3) property is different from the usual (CR_3) on two points. First, as we said above, the problem with usual reducibility candidates for being complete comes from the fact that all neutral normal proof-terms belong to all candidates because of the (CR_3) property. With this new definition, we do not allow neutral normal proof-terms but only neutral not normal proof-terms whose all one-step reducts are in the set. For example, $\alpha\alpha$ can't be added to a set using this new property. In a second step, those neutral not normal proof-terms are not allowed only at the root of a proof-term but at the root of different subtrees of this proof-term. For example, if we note $K = \lambda\alpha.\lambda\beta.\alpha$ and if $\lambda\gamma.\gamma$ is in a set E, then $\lambda\gamma.(K\gamma\ \delta)$ can be added by (CR'_3) to E: we take $\nu = \lambda\gamma.\alpha_1$ and $\mu_1 = K\gamma\ \delta$. μ_1 is neutral, not normal and its only leaf is γ, with $(\gamma/\alpha_1)\nu = \lambda\gamma.\gamma$ which is in

E. We can notice that $\lambda\gamma.(K\gamma\ \delta)$ is not neutral therefore it couldn't has been added with the usual (CR_3) property.

Finally, (CR'_3) allows to add not neutral proof-terms to a set E, unlike (CR_3), but these proof-terms will always reduce to an element of E (still unlike (CR_3)).

4.2 Soundness of $\mathcal{C}'$-models

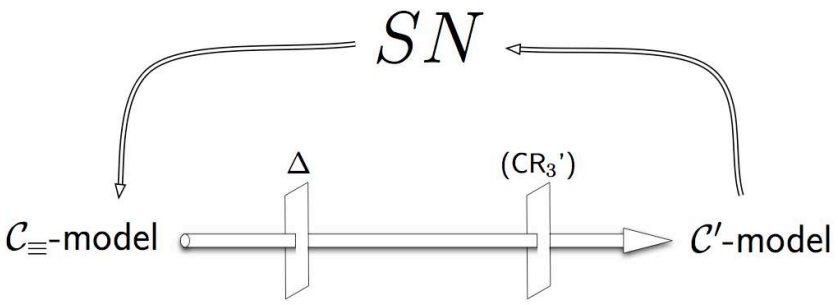
In the usual proof of the adequation lemma, (CR_3) is used twice: first to prove that all candidates are non-empty as they contain all variables, and then to prove that if $\lambda\alpha.\pi$ is of type $A \Rightarrow B$, then it is in $[\![A]\!] \hat{\Rightarrow} [\![B]\!]$. In ($\mathrm{CR}_3$') we have now to suppose explicitely that the considered set is not empty.

In order to prove that $\lambda\alpha.\pi$ is in $[\![A]\!] \hat{\Rightarrow} [\![B]\!]$, we take $\pi' \in [\![A]\!]$ and we prove that $(\lambda\alpha.\pi)\ \pi'$ is in $[\![B]\!]$. We can notice that in this case, $(\lambda\alpha.\pi)\ \pi'$ is not normal and we can therefore use (CR'_3) (with $\nu = \alpha_1$) instead of (CR_3).

Hence, adapting the adequation lemma to (CR'_3) can be done without any difficulty, and therefore, having a $\mathcal{C}'$-model is still a sound condition for strongly normalizing theories in minimal deduction modulo.

4.3 Completeness of $\mathcal{C}'$-models

In order to prove that $\mathcal{C}'$-models provide also a complete semantics for strongly normalizing theories, the idea is to build a morphism (called Cl) from each $\mathcal{C}_{\equiv}$ to this new LDTVA $\mathcal{C}'$ so that each $\mathcal{C}_{\equiv}$-model will be led to a $\mathcal{C}'$-model. Therefore as soon as a theory is strongly normalizing, it has a $\mathcal{C}'$-model: the image by the morphism of the $\mathcal{C}_{\equiv}$-model we built in section 3.



We first filter sets in $\mathcal{C}_{\equiv}$ by a context Δ, in order to consider only proof-terms which are well-typed in a same context. It is convenient that Δ contains an infinite number of variables for each proposition: $\Delta = (\beta_i^A : A)_{A\in\mathcal{P},i\in\mathbb{N}}$. We define the first step of the morphism as:

$Cl^0(E) = \{\pi,\ \exists\Delta' \subseteq \Delta \text{ finite such that } (\Delta', \pi) \in E\}$ (for all sets $E \in \mathcal{C}_{\equiv}$).

Those sets are not in $\mathcal{C}'$ because they do not satisfy (CR'_3) as they contain only well-typed proof-terms. And then we want to saturate those sets by (CR'_3), in

order to obtain elements of $\mathcal{C}'$. Let us now explane why we could not build a morphism from $\mathcal{C}_{\equiv}$ to the usual $\mathcal{C}$. If we write, for this paragraph, $Cl(E)$ for the saturation by (CR_3) of $Cl^0(E)$, we want, for all sets $E, F \in \mathcal{C}_{\equiv}$, to have $Cl(E \mathring{\Rightarrow} F) = Cl(E) \mathring{\Rightarrow} Cl(F)$ (and an analog property for $\mathring{\forall}$ and $\tilde{\forall}$). That is why we allowed not neutral proof-terms in (CR'_3): if we take π in $Cl(F)$ such that $(\Delta, \pi) \notin F$ and α not free in π, then $\lambda\alpha.\pi$ is in $Cl(E) \tilde{\Rightarrow} Cl(F)$ because for all $\pi' \in Cl(E)$, $(\lambda\alpha.\pi)\pi'$ is neutral not normal and all its reducts are in $Cl(F)$ (by induction on the maximal length of a reductions sequence from π': if π' is normal, then the only reduct of $(\lambda\alpha.\pi)\pi'$ is π wich is in $Cl(F)$ by hypothesis). But $\lambda\alpha.\pi \notin Cl(E \mathring{\Rightarrow} F)$ and could not had been added to $Cl(E \mathring{\Rightarrow} F)$ by the usual (CR_3) as it is not neutral.

Finally, the actual definition of $Cl(E)$ is the following one:

$$
\begin{aligned}
Cl^0(E) &= \{\pi,\ \exists \Delta' \subseteq \Delta \text{ finite such that } (\Delta', \pi) \in E\} \\
Cl^{k+1}(E) &= \{\pi, \text{ such that } \exists n \in \mathbb{N}: \\
&\qquad \exists \nu, \exists (\mu_i)_{i \leq n}, \text{ each neutral not normal such that} \\
&\qquad\ \pi = [\mu_i/\alpha_i]_{i \leq n}\ \nu \text{ and } \forall (\rho_i)_{i \leq n}, \text{ s.t. } \forall i \leq n,\ \mu_i \to \rho_i, \\
&\qquad \text{we have } [\rho_i/\alpha_i]_{i \leq n}\ \nu \in Cl^k(E)\} \\
Cl(E) &= \cup_{j \in \mathbb{N}}\, Cl^j(E)
\end{aligned}
$$

We can prove that this function Cl is actually a morphism from each $\mathcal{C}_{\equiv}$ to $\mathcal{C}'$ (in particular, for all $E \in \mathcal{C}_{\equiv}$, $Cl(E)$ satisfies (CR_1) and (CR_2)). Hence, as soon as a theory is strongly normalizing, it has a $\mathcal{C}'$-valued model : $Cl \circ \lfloor . \rfloor_.$. Notice that tt is non-empty because of the fact that $\lfloor . \rfloor_.$ is non-empty. Therefore having a $\mathcal{C}'$-valued model is also a complete condition for strongly normalizing theories.

Conclusion

We have defined a refinement of truth values algebras which allows to build more precise models. Then we exhibited one of these truth values algebras $\mathcal{C}'$ such that having a non-empty $\mathcal{C}'$-valued model is a sound and complete condition for strongly normalizing theories.While soundness is an usual property, this completeness result is, up to our knowledge, the first for strongly normalizing theories, in deduction modulo.

We proved this completeness theorem in an original way: build a structure adapted to the congruence relation and then show that it is also adapted to connectives when the theory is strongly normalizing. This way, we are able to build an interpretation of propositions adapted to the congruence relation, even if the theory is not strongly normalizing.

In future work, we wish to extend this result to other logical frameworks with or without modulo. We want to extend first this result to (complete) Deduction modulo, and to $\lambda\Pi$-calculus modulo [3]. We also want to study how these language-dependent truth values algebras can help us in building models of logical frameworks with dependent types, as $\lambda\Pi$-calculus modulo, or Pure Type Systems.

References

1. A. Church, A formulation of the simple theory of types, *The Journal of Symbolic Logic*, 5:56–68, 1940.
2. D. Cousineau, A completeness theorem for strong normalization in minimal deduction modulo, 2009, (submitted).
3. D. Cousineau and G. Dowek, Embedding Pure Types Systems in the lambda Pi-calculus modulo, *Typed Lambda calculi and Applications*. Lecture Notes in Computer Science 4583, Springer. pp. 102-117. 2007.
4. G.Dowek. Truth values algebras and proof normalization. *Types for proofs and programs*. Lecture Notes in Computer Science 4502. 2007, pp. 110-124.
5. G.Dowek, T.Hardin, and C.Kirchner. Theorem proving modulo. *Journal of Automated Reasoning*, 31:32–72, 2003.
6. G.Dowek, T.Hardin, and C.Kirchner. HOL-lambda-sigma: an intentional first-order expression of higher-order logic. *Mathematical Structures in Computer Science*, 11:1–25, 2001.
7. G.Dowek and A. Miquel, Cut elimination for Zermelo set theory, *manuscript*, 2007.
8. G.Dowek and B.Werner. Proof normalization modulo. *The Journal of Symbolic Logic*, 68(4):1289–1316, 2003.
9. G.Dowek and B.Werner. Arithmetic as a theory modulo. J. Giesel (Ed.), *Term rewriting and applications*, Lecture Notes in Computer Science 3467, Springer-Verlag, 2005, pp. 423-437.
10. M. Fiore, G. Plotkin and D. Turi. Abstract syntax and variable binding. *14th Annual Symposium on Logic in Computer Science*, pages 193–202, 1999.
11. J.-Y. Girard. Une extension de l'interprétation de Gödel à l'analyse, et son application à l'élimination des coupures dans l'analyse et la théorie des types. In J.Fenstad, editor, 2^{nd} *Scandinavian Logic Symposium*, pages 63–92. North Holland, 1971.
12. K.Gödel. Über die Vollständigkeit des Logikkalküls. *Doctoral dissertation*, University Of Vienna. 1929.
13. M. Hamana, Universal Algebra for Termination of Higher-Order Rewriting, *16th International Conference on Rewriting Techniques and Applications*, Lecture Notes in Computer Science 3467, Springer, pp. 135-149, 2005.
14. O.Hermant. A model based cut elimination proof. In 2^{nd} *St-Petersbourg Days in Logic and Computability*, 2003.
15. O.Hermant. *Méthodes sémantiques en déduction modulo*. Doctoral Thesis. Université de Paris 7, 2005.
16. P.A.Melliès, B.Werner. A Generic Normalization Proof for Pure Type Systems. *Types for proofs and programs*. Lecture Notes in Computer Science 1512. 1996.
17. C. Riba. On the Stability by Union of Reducibility Candidates. *10th International Conference on Foundations of Software Science and Computational Structures*, pp 317-331. (2007)
18. W.W. Tait. Intentional interpretations of functionals of finite type I. *The Journal of Symbolic Logic*, 32:198–212, 1967.