The termination of proof reduction in Deduction modulo theory

I. What we have seen so far

Proof, theory, model

Examples of theories

Termination of proof-reduction in Predicate logic

Connecting the notions

Termination of proof-reduction in some theories

Main tool: the notion of model

II. Counter-examples and examples

Counter-examples

As we have seen

$P \longrightarrow P \Rightarrow Q$
$P \longrightarrow Q \wedge \neg P$

consistent but no termination of proof-reduction

Example

$P \longrightarrow Q \Rightarrow Q$

P abbreviation for $Q \Rightarrow Q$
Replace P by $Q \Rightarrow Q$ everywhere

Or direct proof: R_Q set of all strongly terminating proof-terms
$R_P = R_{Q \Rightarrow Q}$
Instead of: R_P set of all strongly terminating proof-terms

Key idea

Same proof as for predicat logic: associate a set R_A of proof-terms to each proposition A
Such that $R_{A \Rightarrow B}$ the set of proof-terms π such that π strongly terminates and when π reduces to $\lambda \alpha : A\ \pi_1$, then for every π' in R_A, $(\pi'/\alpha)\pi_1$ in R_B, etc.

Extra condition: if $A \equiv B$ then $R_A = R_B$

III. Sets candidates to be associated to propositions

Naming operations

$R_{A \wedge B}$ = set of proof-terms π, such that π strongly terminates and if π reduces to $\langle \pi_1, \pi_2 \rangle$ then π_1 in R_A and π_2 in R_B

$E \mathrel{\tilde{\wedge}} F$ = set of proof-terms π, such that π strongly terminates and if π reduces to $\langle \pi_1, \pi_2 \rangle$ then π_1 in E and π_2 in F

Then define $R_{A \wedge B}$ as $R_A \mathrel{\tilde{\wedge}} R_B$

Operations on sets of proof-terms

$\tilde{\wedge}$ maps E and F to the set of proof-terms π, such that π strongly terminates and if π reduces to $\langle \pi_1, \pi_2 \rangle$ then $\pi_1 \in E$ and $\pi_2 \in F$

$\tilde{\Rightarrow}$ maps E and F to the set of proof-terms π, such that π strongly terminates and if π reduces to $\lambda\alpha : A\ \pi_1$ then for every π' in E, $(\pi'/\alpha)\pi_1 \in F$

$\tilde{\top}$, $\tilde{\bot}$, $\tilde{\vee}$, $\tilde{\forall}$, $\tilde{\exists}$

Candidates

The set $\mathcal{C}$ of candidates (reducibility candidates) inductively defined as the smallest set of set of proof terms closed by these operations and intersection

- $\tilde{\top}$ and $\tilde{\bot}$ are candidates
- if E and F are candidates, then $E \mathbin{\tilde{\wedge}} F$, $E \mathbin{\tilde{\vee}} F$, and $E \mathbin{\tilde{\Rightarrow}} F$ are candidates
- if S is a set of candidates, then $\tilde{\forall}\ S$ and $\tilde{\exists}\ S$ are candidates
- if S is a set of candidates, then $\bigcap S$ is a candidate

IV. The algebra of candidates

To prove that proof-reduction terminate in $\equiv$

- Define a function R mapping every proposition A to a candidate R_A s.t.
  1. $R_{A \wedge B} = R_A \tilde{\wedge} R_B$, $R_{A \Rightarrow B} = R_A \tilde{\Rightarrow} R_B$, etc.
  2. if $A \equiv B$, then $R_A = R_B$
- prove all proofs of A are in R_A, hence strongly terminate

Because $R_{A \wedge B} = R_A \tilde{\wedge} R_B$, $R_{A \Rightarrow B} = R_A \tilde{\Rightarrow} R_B$, etc.
Once R defined on atomic propositions, it extends in a unique way

A function mapping atomic propositions to candidates

For each predicate symbol P, a function $\overline{P}$ mapping tuples of terms to candidates: $R_{P(t_1,...,t_n)} = \overline{P}(t_1, ..., t_n)$

In a first step associate, to each term t, an element of an arbitrary set $\mathcal{M}$ and then define a function $\hat{P}$ from $\mathcal{M}^n$ to $\mathcal{C}$

Just a model valued in the algebra $\mathcal{C}$, $R_A = [\![A]\!]$

Condition (2.) rephrases:
if $A \equiv B$ then $[\![A]\!] = [\![B]\!]$

The model is a model of the congruence

The algebra $\mathcal{C}$

Trivial pre-order relation $\leq$ defined by $C \leq C'$ always
operations $\tilde{\top}$, $\tilde{\bot}$, $\tilde{\wedge}$, $\tilde{\vee}$, $\tilde{\Rightarrow}$, $\tilde{\forall}$ and $\tilde{\exists}$ pre-Heyting algebra
(any set is)

Ordered $\subseteq$, complete $(\bigcap)$

Not a Heyting algebra
$\tilde{\top} \neq \tilde{\top} \mathrel{\tilde{\Rightarrow}} \tilde{\top}$
$\lambda\alpha : A\ (\alpha\ \alpha)$ in $\tilde{\top}$ but not in $\tilde{\top} \mathrel{\tilde{\Rightarrow}} \tilde{\top}$

V. The termination of proof-term reduction

If $\mathcal{T}, \equiv$ has a model valued in the algebra $\mathcal{C}$
Then every proof-term this theory strongly terminates

Four easy lemmas

If C is a candidate, then all the elements of C strongly terminate

Let C be a candidate and α a variable, then $\alpha \in C$

If C is a candidate, π is an element of C and $\pi \longrightarrow^* \pi'$, then π' is an element of C

Let C be a candidate and π a proof-term that is an elimination and such that all one-step reducts of π are in C, then π is in C

Main theorem

$\equiv$ a congruence
$\mathcal{M}$ a model valued in the algebra $\mathcal{C}$ of $\equiv$
π a proof-term of type A in a context Γ
θ a substitution mapping variables to terms
ϕ a valuation mapping variables to elements of $\mathcal{M}$
σ a substitution mapping any proof-term variable bound to proposition B in Γ to an element of $[\![B]\!]_\phi$

Then $\sigma\theta\pi$ is an element of $[\![A]\!]_\phi$

Corollaries

If $\mathcal{T}, \equiv$ has a model valued in the algebra $\mathcal{C}$
Then every proof-term this theory strongly terminates

If $\mathcal{T}, \equiv$ is super-consistent
Then, every proof-term this theory strongly terminates

More corollaries

$\varnothing, \equiv$ purely computational and super-consistent

If there exists a proof of A, then there exists one that ends with an introduction rule
If there exists a proof of $\exists x\ A$, then there exists a term t and a proof of $(t/x)A$

And corollaries of corollaries

Every proof-term in Arithmetic strongly terminates
Arithmetic has the witness property

Every proof-term in Simple type theory (with Peano numbers) strongly terminates
Simple type theory (with Peano numbers) has the witness property

VI. Proof-terms reduction in Arithmetic

A class that contains zero that is closed by successor
Two proofs that 100 is in this class

Express these proofs in $HA^{\longrightarrow}$ or in Simple type theory with Peano numbers
Eliminating cuts in one yields the other

Proofs by induction

π proof of $0 \; \epsilon \; c$
π' proof of $\forall x \; (N(x) \Rightarrow x \; \epsilon \; c \Rightarrow S(x) \; \epsilon \; c)$

$\lambda y \; \lambda\alpha \; (\alpha \; c \; \pi \; \pi')$ proof of $\forall y \; (N(y) \Rightarrow y \; \epsilon \; c)$

$$((\lambda y\ \lambda\alpha\ (\alpha\ c\ \pi\ \pi'))\ S^{100}(0)\ \rho_{100})$$

where $\rho_{100} : N(S^{100}(0))$

Second

$$(\pi'\ S^{99}(0)\ \rho_{99}\ (\pi'\ S^{98}(0)\ \rho_{98}\ (\ldots\ (\pi'\ 0\ \rho_0\ \pi))))$$

where $\rho_0 : N(0)$, $\rho_1 : N(S(0))$, $\rho_2 : N(S^2(0))$, etc.

Parigot numbers

$N(S^{100}(0))$ is

$$\forall c\ (0\ \epsilon\ c \Rightarrow \forall x\ (N(x) \Rightarrow x\ \epsilon\ c \Rightarrow S(x)\ \epsilon\ c) \Rightarrow S^{100}(0)\ \epsilon\ c)$$

Only one irreducible proof of this proposition

$$\rho_{100} = \lambda c \lambda x \lambda f\ (f\ S^{99}(0)\ \rho_{99}\ (f\ S^{98}(0)\ \rho_{98}\ (\ldots\ (f\ 0\ \rho_0\ x))))$$

$((\lambda y\ \lambda\alpha\ (\alpha\ c\ \pi\ \pi'))\ S^{100}(0)\ \rho_{100})$
reduces to
$(\rho_{100}\ c\ \pi\ \pi')$
and then to
$(\pi'\ S^{99}(0)\ \rho_{99}\ (\pi'\ S^{98}(0)\ \rho_{98}\ (\ldots\ (\pi'\ 0\ \rho_0\ \pi))))$

Iterator: ρ_{100}, not $S^{100}(0)$

VII. Proofs as programs

π proof of

$$\forall x\ (N(x) \Rightarrow \exists y\ (N(y) \wedge (x = 0 \Rightarrow y = 0) \wedge (\neg x = 0 \Rightarrow y = S(0))))$$

$(\pi\ S^n(0)\ \rho_n)$ proof of

$$\exists y\ (N(y) \wedge (S^n(0) = 0 \Rightarrow y = 0) \wedge (\neg S^n(0) = 0 \Rightarrow y = S(0)))$$

Irreducible form of this proof $\langle t, \langle \pi_1, \pi_2 \rangle \rangle$
where t is a irreducible term expressing a natural number ($S^p(0)$)
π_1 proof of $N(t)$
π_2 proof of

$$(S^n(0) = 0 \Rightarrow t = 0) \wedge (\neg S^n(0) = 0 \Rightarrow t = S(0))$$

Thus if $n = 0$, then $p = 0$ and if $n \neq 0$, then $p = 1$
Computing the function χ: proofs are programs and proof-reduction an interpreter

Provably total computable function

Specification of χ:

$$(x = 0 \Rightarrow y = 0) \wedge (\neg x = 0 \Rightarrow y = S(0))$$

Any computable function f can be specified by a proposition A, s.t. $(n/x, p/y)A$ provable if and only if $p = f(n)$
When

$$\forall x \; (N(x) \Rightarrow \exists y \; (N(y) \wedge A))$$

provable, f provably total in arithmetic

Irreducible form of $(\pi\ S^n(0)\ \rho_n)$: $\langle t, \langle \pi_1, \pi_2 \rangle \rangle$
$t = S^p(0)$ for $p = f(n)$
All functions that are provably total in arithmetic, can be expressed in this programing language

Much more expressive than Simply typed lambda-calculus

$$N(y) \longrightarrow \forall c\ (0\ \epsilon\ c \Rightarrow \forall x\ (N(x) \Rightarrow x\ \epsilon\ c \Rightarrow S(x)\ \epsilon\ c) \Rightarrow y\ \epsilon\ c)$$

permits to type iterators

Next time

Dependent types