

The termination of proof reduction

I. What we have seen so far

Basic notions

Examples of theories

Today: termination of proof reduction

This lecture: in Predicate logic

After the break: in Deduction modulo theory

Extension of the termination proof for simply typed λ -calculus

A new language to express proofs: Brouwer-Heyting-Kolmogorov interpretation of proofs and Curry-de Bruijn-Howard correspondence

II. Proof-terms

Notations for derivation trees

Inductive definition: smallest set closed by some functions

Example: E : smallest set containing $z = 0$ and closed by

$f = x \mapsto x + 2$

n in E if and only if there exists a **derivation** (tree) of E

$$\begin{array}{c} \bar{0} \\ \bar{2} \\ \bar{4} \\ \bar{6} \end{array}$$
$$\begin{array}{c} \bar{z} \\ \bar{f} \\ \bar{f} \\ \bar{f} \end{array}$$
$$\begin{array}{c} \bar{z} \\ \bar{0} \\ \bar{2} \\ \bar{4} \\ \bar{6} \end{array} \begin{array}{c} f \\ f \\ f \\ f \end{array}$$

$$\frac{\frac{}{\vdash \top} \top\text{-intro} \quad \frac{}{\vdash \top} \top\text{-intro}}{\vdash \top \wedge \top} \wedge\text{-intro}$$

Redundant: rule names only

$$\frac{\overline{\top\text{-intro}} \quad \overline{\top\text{-intro}}}{\wedge\text{-intro}}$$

Linear notation for trees: $\wedge\text{-intro}(\top\text{-intro}, \top\text{-intro})$

Shorthand for rule names: $\langle I, I \rangle$

The axiom rule

Natural deduction rules: **functions** from sequents to sequents

For each sequence of propositions A_1, \dots, A_n and each A_i : a different axiom rule

Name of axiom rule parametrized by A_1, \dots, A_n and A_i

$$\frac{\overline{\text{axiom}} \quad \overline{\text{axiom}}}{\wedge\text{-intro}}$$

$$\frac{\overline{\text{axiom}_{P,Q \vdash P}} \quad \overline{\text{axiom}_{P,Q \vdash Q}}}{\wedge\text{-intro}}$$

Contexts

In a sequent: **conclusion** more important than context

Natural deduction rules: deduce propositions from propositions and context recalls which propositions can be used in axiom rule

Name hypotheses: P, Q becomes $\alpha : P, \beta : Q$ and use these names as name of the axiom rule

$$\frac{\bar{\alpha} \quad \bar{\beta}}{\wedge\text{-intro}}$$

$\langle \alpha, \beta \rangle$

Some rules extend context: e.g.

$$\frac{\frac{\pi}{A_1, \dots, A_n, B \vdash C}}{A_1, \dots, A_n \vdash B \Rightarrow C} \Rightarrow\text{-intro}$$

λ shorthand for \Rightarrow -intro: not $\lambda\pi$ or $\lambda B \pi$ but name added hypothesis $\lambda\beta : B \pi$

π proof in context $\alpha_1 : A_1, \dots, \alpha_n : A_n, \beta : B$ and $\lambda\beta : B \pi$ proof in context $\alpha_1 : A_1, \dots, \alpha_n : A_n$

Hypotheses names like variables: β introduced by λ , can be used in π , not elsewhere: bound by λ , scope π

Replacing an hypothesis

$$\frac{\frac{\pi}{\Gamma, A \vdash B} \Rightarrow\text{-intro} \quad \frac{\pi'}{\Gamma \vdash A} \Rightarrow\text{-elim}}{\Gamma \vdash B} \Rightarrow\text{-elim}$$

From π proof of $\Gamma, A \vdash B$, remove the hypothesis A in all sequents, replace the axiom rules on this proposition by π' of $\Gamma \vdash A$

Substitute π' for α (associated to A) in π

A proof of a sequent $A_1, \dots, A_n \vdash B$ expressed as a term in the context $\alpha_1 : A_1, \dots, \alpha_n : A_n$

$$\overline{\Gamma \vdash A_i} \text{ axiom}$$

expressed as α_i

$$\frac{\frac{\pi}{\Gamma \vdash A} \quad \frac{\pi'}{\Gamma \vdash B}}{\Gamma \vdash A \wedge B} \wedge\text{-intro}$$

expressed as $\langle \pi, \pi' \rangle$

$$\frac{\frac{\pi}{\Gamma \vdash A \wedge B}}{\Gamma \vdash A} \wedge\text{-elim}$$

expressed as $\text{fst}(\pi)$

$$\frac{\frac{\pi}{\Gamma \vdash A \wedge B}}{\Gamma \vdash B} \wedge\text{-elim}$$

expressed as $\text{snd}(\pi)$

$$\frac{\frac{\pi}{\Gamma, A \vdash B}}{\Gamma \vdash A \Rightarrow B} \Rightarrow\text{-intro}$$

expressed as $\lambda\alpha : A \pi$

$$\frac{\frac{\pi}{\Gamma \vdash A \Rightarrow B} \quad \frac{\pi'}{\Gamma \vdash A}}{\Gamma \vdash B} \Rightarrow\text{-elim}$$

expressed as $app(\pi, \pi')$ also written $(\pi \pi')$

...

Proof-reduction on terms

$$\frac{\frac{\pi_1}{\Gamma, A \vdash B} \Rightarrow\text{-intro} \quad \frac{\pi_2}{\Gamma \vdash A} \Rightarrow\text{-elim}}{\Gamma \vdash B} \Rightarrow\text{-elim}$$

$((\lambda\alpha : A \pi_1) \pi_2)$ reduces to $(\pi_2/\alpha)\pi_1$

$$((\lambda\alpha : A \pi_1) \pi_2) \longrightarrow (\pi_2/\alpha)\pi_1$$

$$\text{fst}(\langle \pi_1, \pi_2 \rangle) \longrightarrow \pi_1$$

$$\text{snd}(\langle \pi_1, \pi_2 \rangle) \longrightarrow \pi_2$$

The Brouwer-Heyting-Kolmogorov interpretation

So far **just changing notations**: derivations labeled by rule names, a symbol for each rule, from sequents to propositions (variables)

Brouwer-Heyting-Kolmogorov interpretation: a completely **different** approach with **same** result

How do you build a proof of $A \wedge B$?

\wedge -intro rule: build a proof of A and a proof of B

How do you use a proof of $A \wedge B$?

\wedge -elim rules: to build a proof of A or a proof of B

How do you build an ordered pair formed with a proof of A and a proof of B ?

build a proof of A and a proof of B

How do you use an ordered pair formed with a proof of A and a proof of B ?

to build a proof of A or a proof of B

A proof of $A \wedge B$ **built** and **used** like an ordered pair formed with a proof of A and a proof of B

A proof of $A \wedge B$ **is** an ordered pair formed with a proof of A and a proof of B

A proof of $A \Rightarrow B$ is an algorithm mapping proofs of A to proofs of B

$\langle \pi, \pi' \rangle, \lambda x \pi$

and also

- ▶ a proof of \top is always the same object
- ▶ a proof of \perp , there is none
- ▶ a proof of $A \vee B$ is either a proof of A or a proof of B
- ▶ a proof of $\forall x A$ is an algorithm mapping objects t to proofs of $(t/x)A$
- ▶ a proof of $\exists x A$ is an ordered pair formed with an object t and a proof of $(t/x)A$

The Curry-de Bruijn-Howard correspondence

Types for proof-terms

$\Phi(A)$ type of the proofs of A

Proof of $A \Rightarrow B$: algorithm mapping proofs of A to proofs of B

$$\Phi(A \Rightarrow B) = \Phi(A) \rightarrow \Phi(B)$$

Φ isomorphism between propositions and types: the Curry-de Bruijn-Howard correspondence

Identifying isomorphic object

$\lambda\alpha : A \alpha$ has type $A \Rightarrow A$

$\lambda\alpha : A \alpha$ is a proof of $A \Rightarrow A$

$\alpha_1 : A_1, \dots, \alpha_n : A_n \vdash \pi : B$

π is a proof of the sequent $A_1, \dots, A_n \vdash B$

($\alpha_1, \dots, \alpha_n$ names given to the variables standing for proofs of the propositions A_1, \dots, A_n)

Natural deduction rules turned into typing rules

$$\overline{\Gamma \vdash \alpha : A} \text{ axiom if } \alpha : A \in \Gamma$$

$$\frac{\Gamma \vdash \pi : A \quad \Gamma \vdash \pi' : B}{\Gamma \vdash \langle \pi, \pi' \rangle : A \wedge B} \wedge\text{-intro}$$

$$\frac{\Gamma \vdash \pi : A \wedge B}{\Gamma \vdash \text{fst}(\pi) : A} \wedge\text{-elim}$$

$$\frac{\Gamma \vdash \pi : A \wedge B}{\Gamma \vdash \text{snd}(\pi) : B} \wedge\text{-elim}$$

$$\frac{\Gamma, \alpha : A \vdash \pi : B}{\Gamma \vdash \lambda \alpha : A \pi : A \Rightarrow B} \Rightarrow\text{-intro}$$

$$\frac{\Gamma \vdash \pi : A \Rightarrow B \quad \Gamma \vdash \pi' : A}{\Gamma \vdash (\pi \pi') : B} \Rightarrow\text{-elim}$$

$$\frac{\Gamma \vdash \pi : A}{\Gamma \vdash \lambda x \pi : \forall x A} \forall\text{-intro if } x \notin FV(\Gamma)$$

$$\frac{\Gamma \vdash \pi : \forall x A}{\Gamma \vdash (\pi t) : (t/x)A} \forall\text{-elim}$$

Correspondence

$A_1, \dots, A_n \vdash B$ is derivable in Natural deduction

if and only if

there exists a proof-term π such that $\alpha_1 : A_1, \dots, \alpha_n : A_n \vdash \pi : B$
is derivable in this system

Decidability

Unlike Natural deduction
(like typing system for a programming language)

Inductive definition of a **decidable** set (guided by the syntax)

Final rule

π a closed and irreducible proof-term of type A
then it is an introduction, i.e. a term of the form
 $l, \langle \pi_1, \pi_2 \rangle, i(\pi_1), j(\pi_1), \lambda\alpha : B \pi_1, \lambda x \pi_1$, or $\langle t, \pi_1 \rangle$

Corollary: Witness property

III. The termination of proof-term reduction

Follow the termination proof of simply typed λ -calculus

Instead of \rightarrow : \Rightarrow , \top , \perp , \wedge , \vee , \forall , \exists

By induction over A a set of proof-terms R_A

- ▶ If A atomic, then a proof-term is an element of R_A if it strongly terminates
- ▶ A proof-term is an element of $R_{A \Rightarrow B}$ if it strongly terminates and when it reduces to $\lambda\alpha : A \pi_1$ (**introduction**) then for every π' in R_A , $(\pi'/\alpha)\pi_1$ is an element of R_B
- ▶ A proof-term is an element of $R_{A \wedge B}$ if it strongly terminates and when it reduces to $\langle \pi_1, \pi_2 \rangle$ (**introduction**) then π_1 and π_2 are elements of R_A and R_B
- ▶ **etc.**

Easy lemmas

If A proposition and α variable, then $\alpha \in R_A$

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

If A proposition and π proof-term **that is an elimination** such that all one-step reducts of π are in R_A , then π is in R_A

The theorem

π a proof-term of type A in a context Γ

θ a substitution mapping the term-variables to terms of the same sort

σ a substitution mapping proof-term variables bound to a proposition B in Γ to elements of R_B

Then $\sigma\theta\pi$ is an element of R_A

By induction over the structure of π

- ▶ **axiom** $\pi = \alpha, \sigma\theta\alpha = \sigma\alpha$ in R_A

► \Rightarrow -intro $A = B \Rightarrow C$

$\pi = \lambda\alpha : B \rho$ where ρ proof-term of type C

$\sigma\theta\pi = \lambda\alpha : \theta B \sigma\theta\rho$

By induction hypothesis, $\sigma\theta\rho \in R_C$

Reduction sequence issued from $\sigma\theta\pi$ reduces $\sigma\theta\rho$, finite

$\lambda\alpha : \theta B \rho'$ reduct of $\sigma\theta\pi$: ρ' reduct of $\sigma\theta\rho$

τ any proof-term of R_B , $(\tau/\alpha)\rho'$ reduct of $((\tau/\alpha) \circ \sigma)\theta\rho$

By induction hypothesis $((\tau/\alpha) \circ \sigma)\theta\rho$ in R_C

Thus $(\tau/\alpha)\rho' \in R_C$

► \wedge -intro $A = B \wedge C$

$\pi = \langle \rho_1, \rho_2 \rangle$ where ρ_1 of type B and ρ_2 of type C

$\sigma\theta\pi = \langle \sigma\theta\rho_1, \sigma\theta\rho_2 \rangle$

By induction hypothesis, $\sigma\theta\rho_1 \in R_B$ and $\sigma\theta\rho_2 \in R_C$

Reduction sequence issued from $\sigma\theta\pi$ reduces $\sigma\theta\rho_1$ and $\sigma\theta\rho_2$,
finite

$\langle \rho'_1, \rho'_2 \rangle$ reduct of $\sigma\theta\pi$: ρ'_1 is reduct of $\sigma\theta\rho_1$ and ρ'_2 of $\sigma\theta\rho_2$

$\rho'_1 \in R_B$ and $\rho'_2 \in R_C$

► \Rightarrow -elim

$\pi = (\rho_1 \rho_2)$, where ρ_1 of type $B \Rightarrow A$ and ρ_2 of type B
 $\sigma\theta\pi = (\sigma\theta\rho_1 \sigma\theta\rho_2)$

By induction hypothesis $\sigma\theta\rho_1 \in R_{B \Rightarrow A}$ and $\sigma\theta\rho_2 \in R_B$

Termination: n (n') max length seq. issued $\sigma\theta\rho_1$ ($\sigma\theta\rho_2$)

By induction on $n + n'$, $(\sigma\theta\rho_1 \sigma\theta\rho_2) \in R_A$

Only need to prove every of its one step reducts is in R_A

If reduction in $\sigma\theta\rho_1$ or in $\sigma\theta\rho_2$ then induction hypothesis

Otherwise $\sigma\theta\rho_1 = \lambda\alpha \rho'$ reduct is $(\sigma\theta\rho_2/\alpha)\rho'$

By definition of $R_{B \Rightarrow A}$ is in R_A

► \wedge -elim

$\pi = fst(\rho)$ where ρ of type $A \wedge B$

$\sigma\theta\pi = fst(\sigma\theta\rho)$

By induction hypothesis $\sigma\theta\rho \in R_{A \wedge B}$

Termination: n max length seq. issued $\sigma\theta\rho \in R_{A \wedge B}$

By induction on $n, fst(\sigma\theta\rho)$ in R_A

Only need to prove every of its one step reducts is in R_A

If reduction in $\sigma\theta\rho$ then induction hypothesis

Otherwise $\sigma\theta\rho = \langle \rho'_1, \rho'_2 \rangle$ reduct is ρ'_1

By definition of $R_{A \wedge B}$ in R_A

Corollary

Every proof-term in Predicate logic strongly terminates

θ : the substitution mapping each term variable to itself

σ : the substitution mapping each proof-term variables to itself

After the break

The termination of proof-reduction in Deduction modulo theory