型輪講
12 章 Normalization

末永幸平[*]

平成 15 年 5 月 2 日

1　はじめに

今回は, $\lambda^{\rightarrow}$[1] の重要な性質である *normalization* について解説する.
まず初めに例を示す. 以下の 2 つの term を考えよう.

- $(\lambda x.x)(\lambda x.x)$
- $(\lambda x.xx)(\lambda x.xx)$

1 つ目の term は reduction が止まるのに対し, 2 つ目は止まらない. この 2 つの term の間の違いは, 前者が $\lambda^{\rightarrow}$ の範囲で型付けできるのに対し, 後者が型付けできないということである. 実は, 一般に"$\lambda^{\rightarrow}$ の範囲で型付けできる term は, reduction が止まる" ことが示せる. この性質を *Normalization(***正規性***)* という. 今回は, この Normalization の証明を行う.

Normalization は再帰が入った言語では成り立たない. [2] したがって, この性質は実際の言語ではあまり意味をなさないかもしれない. この章の話は, Normalization 自体を理解するというよりは, Normalization のような型システムについての性質を, どうやって証明するかの練習として聞いて頂きたい. [3]

以下では, まず Normalization の定義を行ったあとに, Normalization の証明を行う. 証明の際に, 単純な帰納法を使うとうまくいかないことを示し, それを解決するにはどうすれば良いかを説明する.

---

[*] kohei@yl.is.s.u-tokyo.ac.jp

[1] 実際にこの章で扱うのは, $\lambda^{\rightarrow}$ に基底型の A を加えた calculus である.

[2] `let rec f x = f (x + 1) in f 1` というプログラムは `Int` に型付けできるが, 止まらない. 不動点演算子は $\lambda^{\rightarrow}$ では型付けできないので, 不思議議はないだろう.

[3] 後で出てくる型システム F_ω では, 型自体に $\lambda^{\rightarrow}$ のような計算体系を導入する. このときに, 型のレベルでの Normalization が重要になる.

2 Normalization

まず始めに, normalizability を定義する.

定義: normalizable

> ある term$t$ が normalizable であるとは, $\lambda^{\rightarrow}$ の評価戦略に従って, t から始まる無限の β-reduction の列が存在しないことである.

この定義をふまえて, この章で証明する normalization theorem は次のように書くことができる.

定理: normalization

> $\lambda^{\rightarrow}$ で正しく型付けされた term は, normalizable である.

以下では, Normalization の証明を, 順を追って行う.

3 Normalization の証明

まず初めに, 単純な構造による帰納法がうまく行かないということと, その原因を示す. この問題を解決するために, **帰着可能性**という概念を導入する. 最後に, この概念を用いた証明を行う.

では, 構造による帰納法がうまくいかないことを見てみよう.

証明 (?)

term$t$ の構造に関する帰納法で証明する. t が T_{12} に型付けできたと仮定する.

$t \equiv t_1 t_2$ のときに適用できる型付け規則は

$$\frac{\Gamma \vdash t_1 : T_{11} \rightarrow T_{12} \quad \Gamma \vdash t_2 : T_{11}}{\Gamma \vdash t_1 t_2 : T_{12}} \text{ (T-APP)}$$

である. このとき, 帰納法の仮定により, t_1, t_2 を valuev_1, v_2 に reduction することができる. このとき inversion property により, v_1 を $\lambda x : \sigma.t_1'$ の形に書くことができる.

今までの仮定から $(\lambda x : \sigma.t_1')v_2$ の reduction が有限ステップで停止することを示さなければならない. しかし, これを 1 ステップ reduction すると $[v_2/x]t_1'$

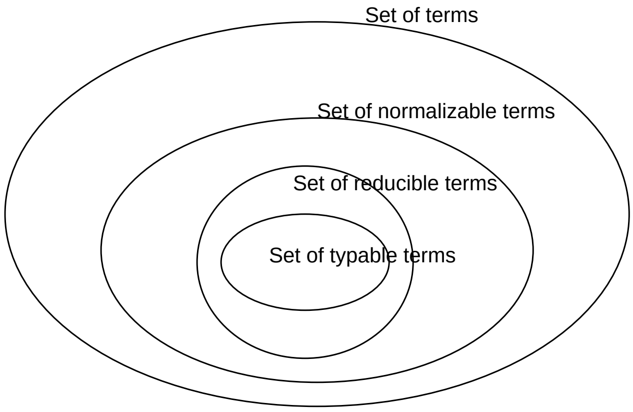

図 1: term, normalizability, 帰着可能性, typability の間の関係.

となる. t'_1 内に x が複数回現れていた場合, $[v_2/x]t'_1$ はもとの $(\lambda x : \sigma.t'_1)v_2$ よりも複雑な構造をとることになり, 帰納法の仮定を使うことができない.

□

このように, 構造による帰納法では normalization の証明は成功しない. これは, T-APP で, t_1 と t_2 が normalizable という帰納法の仮定が, $t_1 t_2$ が normalizable ということを示すのに, 十分強く無かったことが原因である.

このような場合, より強い帰納法の仮定を用いればよい. この際に, 型付け可能な term が全て, この仮定を満たすようにしなければならない. この二つを満たすのが, 以下に述べる R_T (**帰着可能性**) である.

3.1 定義: R_T

型 T について, 型 T を持つ閉じた term を一つ受け取る述語 R_T を, 以下のように帰納的に定義する.

- T が基底型 A の場合, $t : A$ が normalizable ならば, $R_A(t)$ は真.
- $T \equiv T_1 \to T_2$ の場合, $t : T_1 \to T_2$ が normalizable で, かつ任意の $R_{T_1}(s)$ なる s について, $R_{T_2}(ts)$ であれば, $R_{T_1 \to T_2}(t)$ は真.

$R_{T_1 \to T_2}(t)$ であるとき, 単に t が normalizable なだけでなく, どんな $R_{T_1}(t')$ なる t' を適用しても, やはり R_{T_2} を満たすと言っている点で, この述語は normalizability よりも強い条件である. しかしこの述語は, Figure. 1 に示すように, typability よりは弱くなるように作ってある.

この R_T を使って, normalization を次のように証明する. まず, “帰着可能な term $\subset$ normalizable な term” を示す. 次に, “型付け可能な term $\subset$ 帰着可能な term” を示す.

3.2 Lemma: 帰着可能な term $\subset$ normalizable な term

$R_T(t)$ であれば, t は normalizable である.

証明 R_T の定義より明らか.

$\square$

“型付け可能な term $\subset$ 帰着可能な term” の証明に進む前に, 証明で使う lemma を一つ証明する. 帰着可能性が, reduction の前後で保存されるという性質である.

3.3 Lemma

$t : T$ かつ $t \to t'$ であれば, $R_T(t) \Leftrightarrow R_T(t')$.

証明 $T \equiv A$ であれば, t が無限の reduction 列をもたないことと, t' が無限の reduction 列を持たないことは同値なので, 成り立つ. 以下では, $T \equiv T_1 \to T_2$ の場合について証明する.

($\Rightarrow$) t' が無限の reduction 列を持たないことは, t が無限の reduction 列を持たないことから言える.

任意の $R_{T_1}(s)$ なる s を一つ固定する. この時, R_T の定義より, $R_{T_2}(ts)$. $ts \to t's$ なので, 帰納法の仮定より $R_{T_2}(t's)$.

($\Leftarrow$) 同様.

$\square$

続いて, “typable な term $\subset$ 帰着可能な term” を証明する. 証明は型付けの導出に関する帰納法で行う.

証明は大体 straightforward に進むのだが, λ-abstraction の場合にのみ少々工夫する必要がある.

たとえば, $R_{\mathtt{Int}\to\mathtt{Int}}(\lambda x.x+1)$ を示そうとするときに, 帰納法の仮定として $R_{\mathtt{Int}}(x+1)$ を使う必要があるかもしれない. しかし, $R_T(t)$ は閉じた term t に対してしか定義されていないので, このままでは証明できない. このような場合の常套手段として, $R_{\mathtt{Int}}(x+1)$ を, $R_{\mathtt{Int}}(v+1)$ として, 全ての value v に対してこれが成り立つと読み替えて証明する. 一般的に言えば, 自由変数を含

む term を, その自由変数を同じ型の value で置き換えた term の集合に対応させて命題を証明する.

3.4 Lemma: typable な term $\subset$ 帰着可能な term

> $x_1 : T_1, ..., x_n : T_n \vdash t : T$ であるとき, $v_1 : T_1, ..., v_n : T_n$ が $R_{T_i}(v_i)(i = 1...n)$ であるような value であれば, $R_T([x_i := v_i]t)(i = 1...n)$.

証明 (以下の証明では, $x_1 : T_1, ..., x_n : T_n$ を Γ と, $[x_i := v_i]$ を γ と書く. この v_i は任意のものを持ってきたとする.)

$\Gamma \vdash t : T$ の導出に関する帰納法で証明する. 導出に使った最後の規則で場合分けする.

T-VAR **の場合**

$$\frac{T_j = \Gamma(x_j)}{\Gamma \vdash t(\equiv x_j) : T(\equiv T_j)} \text{ T-VAR}$$

証明すべきことは, $R_{T_j}(\gamma(x_j))$ である. ところが, $\gamma(x_j) \equiv v_j$ で, 仮定より $R_{T_j}(v_j)$ なので明らか.

T-ABS **の場合**

$$\frac{\Gamma, x : S_1 \vdash s_2 : S_2}{\Gamma \vdash t(\equiv \lambda x : S_1.s_2) : T(\equiv S_1 \to S_2)} \text{ T-ABS}$$

証明すべきことは, $R_{S_1 \to S_2}(\gamma(\lambda x : S_1.s_2))$ である. これには, $\gamma(\lambda x : S_1.s_2)$ が normalizable であることと, 任意の $R_{S_1}(s)$ なる terms$s$ について, $R_{S_2}((\gamma(\lambda x : S_1.s_2))s)$ であることを言う必要がある.

まず, $\gamma(\lambda x : S_1.s_2)$ が normalizable であることを言う. 一般性を失わずに, 全ての束縛変数は異なるものと仮定できる. [4] したがって,

$$\gamma(\lambda x : S_1.s_2) \equiv \lambda x : S_1.\gamma(s_2).$$

$\lambda x : S_1.\gamma(s_2)$ は value なので normalizable.

次に, 任意の $R_{S_1}(s)$ なる s について, $R_{S_2}((\gamma(\lambda x : S_1.s_2))s)$ となることを示す. $R_{S_1}(s)$ なので, s は normalizable で, $s \to^* v$ となる value$v$ が存在する. 前出の lemma により, この value$v$ も $R_{S_1}(v)$ を満たす. したがって, 帰納法の仮定により, s_2 については $R_{S_2}(\gamma([x := v]s_2))$ が成り立つ.

ところが,

$$\begin{aligned} &(\lambda x : S_1.\gamma(s_2))s \\ \to &(\lambda x : S_1.\gamma(s_2))v \\ \to &\gamma([x := v]s_2) \end{aligned}$$

[4] ことさらに仮定しなくとも, 型環境に入っている変数はすべて異なるとしていたので当然.

なので, やはり前出の lemma により $R_{S_2}((\lambda x:S_1.\gamma(s_2))s)$ が成り立つ. 先に述べた通り

$$\lambda x:S_1.\gamma(s_2) \equiv \gamma(\lambda x:S_1.s_2)$$

なので, $R_{S_2}((\gamma(\lambda x:S_1.s_2))s)$. s は任意なので, これで $R_{S_1\to S_2}([x_i:=v_i]\lambda x:S_1.s_2)$ が言えたことになる.

T-APP **の場合**

$$\frac{\Gamma \vdash t_1 : T_1 \to T_2 \quad \Gamma \vdash t_2 : T_1}{\Gamma \vdash t(\equiv t_1t_2) : T(\equiv T_2)}\ \text{T-APP}$$

証明すべきことは, $R_{T_2}(\gamma(t_1t_2))$ である. 帰納法の仮定により, $R_{T_1\to T_2}(\gamma(t_1))$, $R_{T_1}(gamma(t_2))$. ここで, $R_{T_1\to T_2}$ の定義により[5] $R_{T_2}((\gamma(t_1))(\gamma(t_2)))$. $(\gamma(t_1))(\gamma(t_2)) \equiv \gamma(t_1t_2)$ より, $R_{T_2}(\gamma(t_1t_2))$

□

ここまでの証明により, "typable な term ⊂ 帰着可能な term ⊂ normalizable な term" が言えたので, Normalization が言える. q

3.5 Theorem: Normalization

$\vdash t : T$ ならば t は normalizable.

4 用語に関する注意

Normalizable の定義の頭についている"$\lambda^{\to}$ の評価戦略に従って" という言葉は, 実は要らない. Normalization は, より一般的な full-beta-reduction を持つ体系でも成り立つ. このときには, normalization は *strong normalization(***強正規化性***)* という言葉で refer されることが多い.

[5] normalizability よりも強い仮定を持ってきたのが, ここで効いている.