第 6 章 項の型無し表現

演習 6.1.1. $[\star]$

> $c_0 = \lambda s.\, \lambda z.\, z$;

$c_0 = \lambda.\, \lambda.\, 0$;

> $c_2 = \lambda s.\, \lambda z.\, s\ (s\ z)$;

$c_0 = \lambda.\, \lambda.\, 1\ (1\ 0)$;

> $plus = \lambda m.\, \lambda n.\, \lambda s.\, \lambda z.\, m\ s\ (n\ s\ z)$;

$plus = \lambda.\, \lambda.\, \lambda.\, \lambda.\, 3\ 1\ (2\ 1\ 0)$;

> $fix = \lambda f.\, (\lambda x.\, f\ (\lambda y.\, (x\ x)\ y))(\lambda x.\, f\ (\lambda y.\, (x\ x)\ y))$;

$fix = \lambda.\, (\lambda.\, 1\ (\lambda.\, (1\ 1)\ 0))(\lambda.\, 1\ (\lambda y.\, (1\ 1)\ 0))$;

> $foo = (\lambda x.\, (\lambda x.\, x))(\lambda x.\, x)$;

$foo = (\lambda.\, (\lambda.\, 0))(\lambda.\, 0)$;

- 自己採点: ok

演習 6.1.4. $[\star\star\star\nrightarrow]$

再掲: 定義 6.1.2.

> $\mathcal{T}$ を、以下の条件を満たす集合の最小の族 $\{\mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_2, \dots\}$ とする (1) $0 \le k < n$ ならば $k \in \mathcal{T}_n$ (2) $t_1 \in \mathcal{T}_n$ かつ $n > 0$ ならば $\lambda.\, t_1 \in \mathcal{T}_{n-1}$ (3) $t_1 \in \mathcal{T}_n$ かつ $t_2 \in \mathcal{T}_n$ ならば $(t_1\ t_2) \in \mathcal{T}_n$

構成

n 項の集合 $\mathcal{S} = \{\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2, \dots\}$ を以下のように構成する。

$$\mathcal{S}_n = \bigcup_i \mathcal{S}_n^i \qquad \mathcal{S}_n^0 = \phi$$

$$\begin{aligned} \mathcal{S}_n^{i+1} = \quad & \{0, \dots, n-1\} && \cdots (1') \\ & \cup \{\lambda.\, t_1 \mid t_1 \in \mathcal{S}_{n+1}^i\} && \cdots (2') \\ & \cup \{t_1\ t_2 \mid t_1, t_2 \in \mathcal{S}_n^i\} && \cdots (3') \end{aligned}$$

証明方針

- (a) $\mathcal{S}$ が $\mathcal{T}$ の条件（定義 6.1.2.）を満たすことを示す
- (b) $\mathcal{T}_n$ を満たす任意の集合が $\mathcal{S}_n$ を部分集合として持つ
  （すなわち、$\mathcal{S}_n$ が $\mathcal{T}_n$ の条件を満たす最小の集合である）ことを示す

証明

- (a)
  - (1) (1') より、$\mathcal{S}_n^1 = \{0, \dots, n-1\} \subseteq \mathcal{S}_n$ なので、$0 \le k < n$ ならば $k \in \mathcal{S}_n \in \mathcal{S}$

- (2) $\mathsf{t}_1 \in \mathcal{S}_n$ かつ $n > 0$ ならば、ある i について $\mathsf{t}_1 \in \mathcal{S}_n^i$ とならねばならない。
  このとき (2') より $\lambda.\,\mathsf{t}_1 \in \mathcal{S}_{n-1}^{i+1} \subseteq \mathcal{S}_{n-1} \in \mathcal{S}$
- (3) $\mathsf{t}_1 \in \mathcal{S}_n$ かつ $\mathsf{t}_2 \in \mathcal{S}_n$ ならば、ある i について $\mathsf{t}_1, \mathsf{t}_2 \in \mathcal{S}_n^i$ とならねばならない。
  このとき (3') より $(\mathsf{t}_1\ \mathsf{t}_2) \in \mathcal{S}_n^{i+1} \subseteq \mathcal{S}_n \in \mathcal{S}$

- (b)
  - 方針：ある $\mathcal{S}'_n$ が定義 6.1.2. を満たすと仮定する。i に関する完全帰納法を用いて、すべての i について $\mathcal{S}_n^i \subseteq \mathcal{S}'_n$ を示す。これから $\mathcal{S}_n \subseteq \mathcal{S}'_n$ が得られるのは明らか。
  - すべての $j < i$ について、$\mathcal{S}_n^j \subseteq \mathcal{S}'_n$ を仮定し、$\mathcal{S}_n^i \subseteq \mathcal{S}'_n$ を示す。
    - $i = 0$ のとき、$\mathcal{S}_n^i = \phi \subseteq \mathcal{S}'_n$
    - $i > 0$ のとき、$i = j + 1$ なる j が存在して、$\mathsf{t} \in \mathcal{S}_n^{j+1}$ とする。
      - t が定数ならば、(1) より $\mathsf{t} \in \mathcal{S}'_n$
      - t がある $\mathsf{t}_1 \in \mathcal{S}_{n+1}^j$ に対して $\lambda.\,\mathsf{t}_1$ の形ならば、$\mathsf{t}_1 \in \mathcal{S}'_{n+1}$ であり、かつ、$n + 1 > 0$ なので (2) より $\lambda.\,\mathsf{t}_1 \in \mathcal{S}'_n$
      - t がある $\mathsf{t}_1, \mathsf{t}_2 \in \mathcal{S}_n^j$ に対して $\mathsf{t}_1\ \mathsf{t}_2$ の形ならば、$\mathsf{t}_1, \mathsf{t}_2 \in \mathcal{S}'_n$ なので、(3) より $(\mathsf{t}_1\ \mathsf{t}_2) \in \mathcal{S}'_n$
  - よって、すべての i について $\mathcal{S}_n^i \subseteq \mathcal{S}'_n$ を証明した。$\mathcal{S}_n$ は $\mathcal{S}_n^i$ の和集合なので、$\mathcal{S}_n \subseteq \mathcal{S}'_n$ が得られる。

演習 6.1.5. $\left[\text{推 奨}, \star\star\star\right]$

(1) $removenames_\Gamma(\mathsf{t})$を定義せよ

$$
\begin{aligned}
& & removenames_\Gamma(\mathsf{t}) &= rem_\Gamma(\mathsf{t}, 0, \phi) \\
& & rem_\Gamma(\mathsf{x}, d, D) &= \begin{cases} d + \Gamma(\mathsf{x}) & \mathsf{x} \in dom(\Gamma) \\ d - 1 - D(\mathsf{x}) & otherwise \end{cases} \\
where & & rem_\Gamma(\lambda\mathsf{x}.\,\mathsf{t}_1, d, D) &= \lambda.\, rem_\Gamma(\mathsf{t}_1, d + 1, D \cup \{\mathsf{x} \mapsto d\}) \\
& & rem_\Gamma(\mathsf{t}_1\ \mathsf{t}_2, d, D) &= rem_\Gamma(\mathsf{t}_1, d, D)\ rem_\Gamma(\mathsf{t}_2, d, D)
\end{aligned}
$$

(2) $restorenames_\Gamma(\mathsf{t})$を定義せよ

- あるマッピングの集合 A に対し A^{-1} のように書いたとき、マッピングの方向が逆になった集合を表すとする
  - 例: $\{x \mapsto 4\}^{-1} = \{4 \mapsto x\}$

$$
\begin{aligned}
& & restorenames_\Gamma(\mathsf{t}) &= res_\Gamma(\mathsf{t}, 0, \phi) \\
& & res_\Gamma(n, d, D) &= \begin{cases} \Gamma^{-1}(n - d) & n - d \in dom(\Gamma^{-1}) \\ D^{-1}(d - 1 - n) & otherwise \end{cases} \\
where & & res_\Gamma(\lambda.\,\mathsf{t}_1, d, D) &= \lambda\mathsf{x}_{fresh}.\, res_\Gamma(\mathsf{t}_1, d + 1, D \cup \{\mathsf{x}_{fresh} \mapsto d\}) \\
& & & \quad where \quad \mathsf{x}_{fresh} = fresh(\mathcal{V} \setminus (\Gamma \cup D)) \\
& & & \quad where \quad fresh(X) = X^{-1}(\min dom(X^{-1})) \\
& & res_\Gamma(\mathsf{t}_1\ \mathsf{t}_2, d, D) &= res_\Gamma(\mathsf{t}_1, d, D)\ res_\Gamma(\mathsf{t}_2, d, D)
\end{aligned}
$$

- 自己採点: わかりません（最も右にある x のインデックスを求めるのに、再帰的定義だと現在の深さ d とラムダ抽象 λx. に付随する変項 x が出現したときの深さを保持する集合 D が必要なんじゃない？というアイディアで書きました）

演習 6.2.2. $\left[\star\right]$

省略

演習 6.2.3. $\left[\star\star\ \not\to\right]$

- 任意の t, d, c に対して $\begin{cases} \uparrow_c^d (\mathsf{t}) \leq n+d \text{ 項} \\ \uparrow_c^d (\mathsf{t}) \geq 0 \text{ 項} \end{cases}$ であることを示したい
- とりあえず境界部分のケースを考える
  - $n+d$ 項となる場合:
    $\mathsf{t} = 0\ 1\ \ldots (n-d-1) \underbrace{(\lambda. \ldots (\lambda.}_{d-1} (n-d)\ \ldots\ (n-1)) \underbrace{\ldots)}_{d-1} (n-d)\ \ldots\ (n-1) \quad (c = n-d-1, 0 \leq d \leq n-1)$
    - 実際にシフトすると:
      $\uparrow_{n-d-1}^{d} (\mathsf{t}) = 0\ 1\ \ldots (n-d-1) \underbrace{(\lambda. \ldots (\lambda.}_{d-1} (n-d)\ \ldots\ (n-1)) \underbrace{\ldots)}_{d-1} n\ \ldots\ (n-1+d)$
  - 0 項となる場合: $\mathsf{t} = 0$
    - 0 項であり n 項でもあるはず...
- $n+d+1$ 項以上は作り得ないことを示すには $n+d$ 以上のインデックスを作り得ないことを示せばよい
  - 任意の項で $\uparrow_c^d$ なるシフトで内部的に $d+1$ 以上シフトされるインデックスが存在し得ないことを示す
  - シフトの定義から自明と言える?

演習 6.2.5. $\left[\star\right]$

省略

演習 6.2.6. $\left[\star\star\ \not\to\right]$

[TODO]

演習 6.2.7. $\left[\star\ \not\to\right]$

省略（ちゃんと書きましたよ...?）

演習 6.2.8. $\left[\text{推 奨}, \star\star\star\right]$

(1) どのような定理が証明される必要があるか

通常の項に対して代入を行ってから名前を除去したときと、名前を除去してから代入を行ったときで同じ項が得られる

(2) 証明せよ

[TODO]

演習 6.3.1 $\left[\star\right]$

- t_{12} と v_2 が負のインデックスを含まない限り、負のインデックスを含む項が作られるおそれはない。

E-AppAbs では、t_{12} 中の 0 の出現が全て $\uparrow^1 (\mathsf{v}_2)$ に置き換わる。そのため、$\uparrow^1 (\mathsf{v}_2)$ の自由変数が全て 1 以上ならば、t_{12} 中の変数はすべて 1 以上である。

また、v_2 中の 0 以上の自由変数は $\uparrow^1 (\mathsf{v}_2)$ においてすべて 1 以上となっている。

以上から、$[0 \mapsto \uparrow^1 (\mathsf{v}_2)]\mathsf{t}_{12}$ の自由変数はすべて 1 以上であり、E-AppAbs によって負のインデックスを含む項は出現しない。

演習 6.3.2 $[\star\star\star]$

De Bruijn レベルを、通常の項 t に対する名無し項構成関数 $removenames'_\Gamma(\mathsf{t})$ を構成することによって定義する。

$$
\begin{aligned}
& removenames'_\Gamma(\mathsf{t}) = rem'_\Gamma(\mathsf{t}, 0, \phi) \\
& rem'_\Gamma(\mathsf{x}, d, D) = \begin{cases} d + \Gamma(\mathsf{x}) & \mathsf{x} \in dom(\Gamma) \\ D(\mathsf{x}) & otherwise \end{cases} \\
where \quad & rem'_\Gamma(\lambda \mathsf{x}.\, \mathsf{t}_1, d, D) = \lambda.\, rem'_\Gamma(\mathsf{t}_1, d+1, D \cup \{\mathsf{x} \mapsto d\}) \\
& rem'_\Gamma(\mathsf{t}_1\ \mathsf{t}_2, d, D) = rem'_\Gamma(\mathsf{t}_1, d, D)\ rem'_\Gamma(\mathsf{t}_2, d, D)
\end{aligned}
$$

変化があるのは実質 x がマッチした場合の第2項だけであり、$d - 1 - D(\mathsf{x})$ が $D(\mathsf{x})$ に変化している。

つまり、束縛変数のレベルはそのままラムダ項として出現したときのラムダの深さとなる。

同様に名無し項 t に対する通常の項構成関数 $restorenames'_\Gamma(\mathsf{t})$ を構成する。

[TODO]

演習 6.1.5. と同様、2つの関数が逆写像（のような）性質を持つことができれば、通常の項を経由してインデックスとレベルは相互に復元できるため、インデックスとレベルは同型となる。

2つの関数が逆写像（のような）性質を持つことに関する証明は [TODO]

- ここまで書いておいて思ったが、逆写像性を言えたからといって変換の一意性があるわけではないから、通常の項を経由する方法はうまく行かないかも…