

第 13 章 参照 (後半)

テキストの解答要約はこんな感じで引用表現にする(引用じゃないけど)

演習 13.4.1. [★]

- 循環的なストアを具体的に作れるか？

```
a = ref (λx:Nat. 999);
b = ref (λx:Nat. (!a) x);
a := λx:Nat. (!b) x
```

= は let 式中じゃないと書けない(記憶喪失)

- 定義 13.5.1. ストア μ に対する well-typed $\Gamma \mid \Sigma \vdash \mu$ の定義

演習 13.5.2. [★★]

- well-typed でかつ、複数のストア片付けを用いる場合はあるか？

とれない。

とれる。自己参照みたいな形にすると $\text{Unit} \rightarrow \text{Unit}$ かもしれないし $\text{Unit} \rightarrow \text{Unit} \rightarrow \text{Unit}$ かもしれない、みたいなパターンが作れる。

- 定理 13.5.3. [保存]
 - 補題 13.5.4. [代入]
 - 補題 13.5.5. ストア内のセルの中身を適当な型を持つ新しい値で置き換えてもストアの型は変わらない
 - 補題 13.5.6. 弱化補題(より大きいストア片付けを持っても項の well-typed 性は守られる)
 - 上記の補題と逆転補題を使って、評価導出についての帰納法で示せる。
- 13.5.7. [進行]

演習 13.5.8. [推奨, ★ ★ ★]

- well-typed 項に対して正規化可能性を持つか？
 - 正規化可能: well-typed 項が有限ステップで停止する(12章より)

持つ。単純型付きラムダ計算に Unit と参照を加えた計算体系 $\rightarrow \text{Unit Ref}$ では、 $\vdash t : T$ ならば t は正規化可能である。

[証明]

- 概略(定理 12.1.6. [正規化] を本体系に合わせて拡張する)
 - $\vdash t : T$ ならば $R_T(t)$ である(補題 12.1.5 の拡張より)
 - R_T のことを 飽和集合 または 簡約可能候補 と呼ぶこともある
 - $R_T(t)$ ならば t は正規化可能である(補題 12.1.3 の拡張より)
- 必要な作業

- \rightarrow Unit Ref のための定義 12.1.2. における $R_T(t)$ の拡張
 - これによって、補題 12.1.3 の文言は変えずに再利用できる
- 補題 12.1.5 の拡張
 - 内部で補題 12.1.3 の拡張と補題 12.1.4 の拡張が必要
- 定義 12.1.2. の拡張(もしかして、ストア型付けも考慮したほうが良い?)
 - $R_{\text{Unit}}(t)$ であるのは、 t が停止するとき、かつそのときのみである。
 - $R_{T_1 \rightarrow T_2}(t)$ であるのは、 t が停止し、さらに、 $R_{T_1}(s)$ ならば常に $R_{T_2}(t\ s)$ であるとき、かつそのときのみである。
 - $R_{\text{Ref } T}(t)$ であるのは、 t が停止し、さらに、ある $t = \text{ref } s$ となる s が存在して $R_T(s)$ であるとき、かつそのときのみである。
- 補題 12.1.4 の拡張
 - 型 T の構造に関する帰納法による
 - $T = \text{Unit}$ のとき、は1ステップ評価とかがないので終わり
 - $T = \text{Ref } T_1$ のとき、
 - 十分性 (\implies)
 - $R_{\text{Ref } T_1}(t)$ かつ、ある $t = \text{ref } s$ となる s が存在すると仮定する。定義より $R_{T_1}(s)$ である。しかし、 $t \rightarrow t'$ であるから、ある $t' = \text{ref } s'$ となる s' が存在して、型 T_1 に関する仮定より $R_{T_1}(s')$ となる。これは任意の s について成り立つから、 $R_{\text{Ref } T_1}(t')$ が得られる。
 - 必要性 (\impliedby) についても同様である。
- 補題 12.1.5 の拡張
 - T-Unit の場合、直ちに明らか。
 - T-Loc の場合、T-Varみたいな感じ
 - $T = T_1 = \Sigma(l)$ 直ちに明らか
 - T-Ref の場合、T-Absみたいな感じ
 - (ここで力尽きた)
 - T-Deref の場合、
 - T-Assign の場合、T-Appみたいな感じ

正しく型付けされていて正規化可能性を持たない項が存在する。

関数型への参照セルと、関数の本体を作り、相互に呼び出すように参照を貼り直すと、再帰が実現する。