Types and Programming Languages
Chapter 8,9

Rei Tomori

July 11, 2025

概要

今回は以下の内容を扱う:

1. §8. 型付き算術式
2. §9. STLC

§8. 型付き算術式

算術式の構文は

$$\mathtt{t} ::= \mathtt{true} | \mathtt{false} | \mathtt{if\ t\ then\ t\ else\ t} | \mathtt{0} | \mathtt{succ\ t} | \mathtt{pred\ t} | \mathtt{iszero\ t}$$

により定義され，各項を評価すると結果は行き詰まり項もしくは値，つまりブール値 $\mathtt{v} ::= \mathtt{true} | \mathtt{false}$ または数値 $\mathtt{nv} ::= \mathtt{0} | \mathtt{succ\ nv}$ になる．

ある項が行き詰まり項の状態に至ることを，その項を実際に評価せずにいいたい．そのためには，数値に評価される項とブール値に評価される項を区別し，各コンストラクタが引数として取れる項を限定する必要がある．そのために，型 $\mathrm{Nat}, \mathrm{Bool}$ を導入する．

記法
S, T, U は型の上を動くメタ変数とする．また，項 t が型 T を持つというとき，t を評価した結果が明らかに，i.e. 静的に型 T の値になることを意味する．

算術式のための型付け関係を $t : T$ と書き，以下の推論規則によって定める．

B のための型付け規則

新しい構文形式

$$T ::= \mathtt{Bool}$$

新しい型付け規則

$$\frac{}{\mathtt{true} : \mathtt{Bool}}\ \text{T-TRUE}$$

$$\frac{}{\mathtt{false} : \mathtt{Bool}}\ \text{T-FALSE}$$

$$\frac{t_1 : \mathtt{Bool} \qquad t_2 : T \qquad t_3 : T}{\mathtt{if}\ t_1\ \mathtt{then}\ t_2\ \mathtt{else}\ t_3 : T}\ \text{T-IF}$$

NB のための型付け規則

新しい構文形式

$$T ::= \ldots \mathtt{Nat}$$

新しい型付け規則

$$\frac{}{\mathtt{0} : \mathtt{Nat}}\ \text{T-ZERO}$$

$$\frac{t_1 : \mathtt{Nat}}{\mathtt{succ}\ t_1 : \mathtt{Nat}}\ \text{T-SUCC}$$

$$\frac{t_1 : \mathtt{Nat}}{\mathtt{pred}\ t_1 : \mathtt{Nat}}\ \text{T-PRED}$$

$$\frac{t_1 : \mathtt{Nat}}{\mathtt{iszero}\ t_1 : \mathtt{Bool}}\ \text{T-ISZERO}$$

Def.8.2.1.

算術式のための型付け関係は p.19 における規則の全てのインスタンスを満たす，項と型の最小の二項関係である．項 t が型付け可能 (正しく型付けされている) とは，$\exists T, t : T$ なること．

たとえは $\mathtt{succ}\ t_1$ が型付け可能のとき，その型は Nat かつ $t_1 : \mathtt{Nat}$ である．このような性質は逆転補題 (生成補題) と呼ばれる．

Lem8.2.2. 型付け関係の逆転.

1. $\mathtt{true} : R$ ならば $R = \mathtt{Bool}$.
2. $\mathtt{false} : R$ ならば $R = \mathtt{Bool}$.
3. $\mathtt{if}\ t_1\ \mathtt{then}\ t_2\ \mathtt{else}\ t_3 : R$ ならば $t_1 : \mathtt{Bool} \land t_2 : R \land t_3 : R$.
4. $0 : R$ ならば $R = \mathtt{Nat}$.
5. $\mathtt{succ}\ t_1 : R$ ならば $R = \mathtt{Nat} \land t_1 : \mathtt{Nat}$.
6. $\mathtt{pred}\ t_1 : R$ ならば $R = \mathtt{Nat} \land t_1 : \mathtt{Nat}$.
7. $\mathtt{iszero}\ t_1 : R$ ならば $R = \mathtt{Nat} \land t_1 : \mathtt{Nat}$.

演習 8.2.3

問題
正しく型付けされた項の全ての部分項は正しく型付けされている.

Proof.
項 $\mathtt{t}$ の帰納法.

1. $\mathtt{t}$ が定数のとき. $\mathtt{t}$ の任意の部分項は $\mathtt{t}$ 自身に限られ, p.19 の規則から全て well-typed なことより従う.
2. 各項 $\mathtt{t}$ に対し, well-typed な $\mathtt{t}$ の直接の部分項たち $\mathtt{t}'$ の任意の部分項 $\mathtt{s}$ は well-typed なことを仮定する.
   1. $\mathtt{t}$ が $\mathtt{t}_1$ に $\mathtt{succ}, \mathtt{pred}, \mathtt{iszero}$ のいずれかを適用した項のとき. $\mathtt{t}$ が well-typed なことを仮定する. 逆転補題により, $\mathtt{t}_1 : \mathtt{Nat}$ より $\mathtt{t}_1$ は well-typed である. 帰納法の仮定より, $\mathtt{t}_1$ の任意の部分項は well-typed, かつ $\mathtt{t}$ はそれ自身の部分項なので示される.
   2. $\mathtt{t} = \mathtt{if}\ \mathtt{t}_1\ \mathtt{then}\ \mathtt{t}_2\ \mathtt{else}\ \mathtt{t}_3$ のとき. $\mathtt{t}$ が well-typed なことを仮定すると, 逆転補題より $\mathtt{t}_1, \mathtt{t}_2, \mathtt{t}_3$ は well-typed. 帰納法の仮定より, (1) と全く同様にして主張が従う.

□

型付け導出とその一意性.

Def. 型付け導出

型付け関係 t : T の型付け導出は，t : T を結論とする型付け規則のインスタンスの木である.

以下の性質は一般には成りたたない．たとえば subtyping などを持つ型体系においては，ある項が存在して，型付け可能かつ 2 通り以上の型をもちうる.

Thm.8.2.4

各項 t は高々一つの型を持つ．i.e.t が型付け可能ならばその型は一意的である.

§8.3. 安全性 = 進行 + 保存

p.19 で定めた型体系の安全性[1]，つまり任意の項 t に対し，t が型付け可能ならば t の評価は行き詰まらないことを示す．以下の 2 つの補題を示すことによって証明しよう．

Thm.8.3.2. 進行定理

$$\forall t, \exists T, t : T \Rightarrow (t\text{は値} \vee \exists t', t \to t')$$

Thm.8.3.3. 保存定理

$$\forall t, t', \forall T, t : T \wedge t \to t' \Rightarrow t' : T$$

[1]つまりは健全性

証明の準備

Bool 型と Nat 型の標準形，i.e. それらの型を持つ正しく型付けされた値について，以下の性質が成り立つ．

Lem.8.3.1. 標準形

1. v が Bool 型の値ならば，v は true または false．
2. v が Nat 型の値ならば，v は nv ::= 0|succ nv の定める文法による数値．

Proof.
(1) ブール値の文法 v ::= true|false と併せると，この言語における値は 4 つの形 true, false, 0, succ nv(ただし nv は数値) を持つ．はじめの 2 つの場合は **B** の型付け規則より即座に成り立つ．後者 2 つについては，逆転補題によりともに Bool = Nat が導かれ，これは不可能．

(2) はじめ 2 つの場合は逆転補題より Nat = Bool が導かれ，これは不可能．後者 2 つに対しては即座に成り立つ． □

Thm.8.3.2. の証明

$\mathtt{t} : \mathsf{T}$ の導出に関する帰納法で以下の定理を示す．

Thm.8.3.2. 進行定理

$$\forall \mathtt{t}, \exists \mathsf{T}, \mathtt{t} : \mathsf{T} \Rightarrow (\mathtt{t}\text{は値} \vee \exists \mathtt{t}', \mathtt{t} \to \mathtt{t}')$$

Proof.

1. T-TRUE，T-FALSE，T-ZERO の場合は $\mathtt{t}$ は値より従う．
2. それ以外の場合．$\mathtt{t} : \mathsf{T}$ の直接の部分導出に対して主張を仮定．
   1. T-IF の場合．$\mathtt{t} = \mathtt{if}\ \mathtt{t}_1\ \mathtt{then}\ \mathtt{t}_2\ \mathtt{else}\ \mathtt{t}_3 : \mathsf{T}$ とする．逆転補題から $\mathtt{t}_1 : \mathtt{Bool}, \mathtt{t}_2 : \mathsf{T}, \mathtt{t}_3 : \mathsf{T}$．帰納法の仮定より $\mathtt{t}_1$ は値または $\exists \mathtt{t}_1', \mathtt{t}_1 \to \mathtt{t}_1'$．$\mathtt{t}_1$ が値のとき，Lem.8.3.1 より $\mathtt{t}_1 = \mathtt{true}$ または $\mathtt{t}_1 = \mathtt{false}$．このとき E-TRUE(FALSE) が適用できる．そうでない場合は E-IF が適用される．
   2. T-SUCC(PRED|ISZERO) の場合．$\mathtt{t} = \mathtt{succ}\ \mathtt{t}_1$ の場合を考えればよい．逆転補題より $\mathtt{t}_1 : \mathtt{Nat}$．帰納法の仮定より $\mathtt{t}_1$ は値または $\exists \mathtt{t}_1', \mathtt{t}_1 \to \mathtt{t}_1'$．値のときは Lem.8.3.1 より $\mathtt{t}_1$ は数値で，$\mathtt{t}$ も数値．1 ステップの評価が可能な場合は E-SUCC が適用され，$\mathtt{t} \to \mathtt{succ}\ \mathtt{t}_1'$．

□

Thm.8.3.3. の証明

$\mathtt{t} : \mathtt{T}$ の導出に関する帰納法で以下の定理を示す．$\mathtt{t} \to \mathtt{t}'$ の評価の導出に関する induction による証明は Ex.8.3.4 で行なう．

Thm.8.3.3. 保存定理

$$\forall \mathtt{t}, \mathtt{t}'. \forall T. \mathtt{t} : \mathtt{T} \land \mathtt{t} \to \mathtt{t}' \Rightarrow \mathtt{t}' : \mathtt{T}$$

Thm.8.3.3. の証明

Proof.
帰納法の各ステップにおいて，任意の部分導出に対して題意を仮定する．
i.e.$\mathsf{s} : \mathsf{S}$ が部分導出で証明されるとき，$\forall \mathsf{s}'.\mathsf{s} : \mathsf{S} \wedge \mathsf{s} \to \mathsf{s}' \Rightarrow \mathsf{s}' : \mathsf{S}$ を仮定する．最後の導出で使われた型付け規則で分ける．

1. T-(TRUE|FALSE|ZERO) のとき．規則の形より $\mathsf{t} = \mathtt{true}$，かつ $\mathsf{T} : \mathtt{Bool}$ なることが分かる．しかし t は正規形より $\forall \mathsf{t}', \neg(\mathsf{t} \to \mathsf{t}')$ なので，主張は成立．他の規則も同様．
2. T-IF のとき．規則の形より $\mathsf{t} = \mathtt{if}\ \mathsf{t}_1\ \mathtt{then}\ \mathsf{t}_2\ \mathtt{else}\ \mathsf{t}_3 : \mathsf{T}, \mathsf{t}_1 : \mathtt{Bool}, \mathsf{t}_2 : \mathsf{T}, \mathsf{t}_3 : \mathsf{T}$ なる項 $\mathsf{t}_1, \mathsf{t}_2, \mathsf{t}_3$ および型 T を取る．帰納法の仮定より，$\mathsf{t}_1 : \mathtt{Bool}, \mathsf{t}_2 : \mathsf{T}, \mathsf{t}_3 : \mathsf{T}$ を結論とする部分導出が存在する．$\mathsf{t} \to \mathsf{t}'$ を仮定すると，規則は E-IF(TRUE|FALSE)，E-IF のいずれかが適用できる．
   - E-IF(TRUE|FALSE) のとき．前者の場合を考える．規則の形より $\mathsf{t}_1 = \mathtt{true}$ で，結果の項は $\mathsf{t}' = \mathsf{t}_2 : \mathsf{T}$ より従う．$\mathsf{t}_1 = \mathtt{false}$ の場合も同様．
   - E-IF のとき．規則の形より $\mathsf{t}_1 \to \mathsf{t}_1'$ なる t_1' が存在する．帰納法の仮定より $\mathsf{t}_1 : \mathtt{Bool}$ を併せると $\mathsf{t}_1' : \mathtt{Bool}$．よって規則 T-IF を使うと $\mathsf{t}' = \mathtt{if}\ \mathsf{t}_1'\ \mathtt{then}\ \mathsf{t}_2\ \mathtt{else}\ \mathsf{t}_3 : \mathsf{T}$ より正しい．
3. T-(SUCC|PRED|ISZERO) のときは (2) と同様．

□

Thm.8.3.3 の別証

Proof.
評価導出による証明．帰納法の各ステップにおいて，任意の部分導出に対して題意を仮定する．つまり，$s \to s'$ が直接の部分導出で証明できるとき，$\forall S, s : S \land s \to s' \Rightarrow s' : S$ を仮定し，最後に使われた規則で分ける．詳細は白板で説明する． □

Ex.8.3.6: 主部展開

Ex.8.3.6
$\mathtt{t} \to \mathtt{t}'$ かつ $\mathtt{t}' : \mathtt{T}$ ならば $\mathtt{t} : \mathtt{T}$ は常に成り立つだろうか．成り立つなら証明し，そうでないなら反例を示せ．

証明．
反例としては，$\mathtt{t} = \mathtt{if\ true\ then\ 0\ else\ false}$ なる項が考えられる，
$\mathtt{t} \to \mathtt{0}$ かつ $\mathtt{0} : \mathtt{Nat}$ であるが，$\mathtt{t}$ は型付けができない． □

Ex.8.3.7
評価関係が演習 3.5.17 の大ステップスタイルで定義されていると仮定せよ．
このとき直感的な型安全性はどのように形式化すべきか．

解答.
進行定理は次のように書ける．小ステップのときに比べ，型付けされた項の評価の停止性を仮定することになる: $\forall t.\exists T.\exists v.t : T \Rightarrow t \Downarrow v$.

保存定理は，$\forall t.\forall v.\forall T.t : T \land t \Downarrow v \Rightarrow v : T$ □

Ex.8.3.8.
演習 3.5.16 のように無意味な項を wrong という明示的な状態へ簡約する規則を追加した評価関係を仮定せよ．ここで型安全性はどのように形式化すべきか.

Proof.
進行定理は次のように書ける: 任意の値でない項 t に対して，t は wrong または $t' \neq t, \mathtt{wrong}$ なる t' に評価される.

一方，wrong はいかなる型も持ちえないことに注意すると，保存定理は次のように書ける:$\forall t, t'.\forall T. t : T \wedge t \to t' \Rightarrow t' \neq \mathtt{wrong} \wedge t' : T.$ □

§9. 単純型付きラムダ計算

- 純粋な λ 計算のプリミティブとブール式を組合せた言語に対して，以下の条件を満たす型システムを構築する:
  - 型安全性を満たす, i.e. 進行と保存定理を満たす.
  - 保守的すぎない.
- 純粋な λ 計算は Turing 完全なので，とくに発散するプログラムを構成できる $(\mathsf{ex}.(\lambda \mathsf{x}.\mathsf{x}\ \mathsf{x})(\lambda \mathsf{x}.\mathsf{x}\ \mathsf{x}))$ ので，正確な型解析が出来るわけではない
  - 型付け可能なプログラムには制約が課される
- 以下，ブール型のみを基底型として持ち，型構築子として関数型を定める $(\to)$ のみを持つ型体系を持つ $\mathsf{STLC}(\lambda_{\to})$ について考察するものとする.

関数に型付けする場合，各抽象に対し関数を表わす型を 1 つ定めるだけでは，たとえば $\lambda x.\mathtt{true}$ と $\lambda x.\lambda y.y$ といった項を区別できず，保守的すぎる．そこで，関数の型をその引数と返値で区別する．

Def.9.1.1.(単純型の集合)

型 $\mathtt{Bool}$ 上の単純型の集合は次の文法により生成される．

$$\mathsf{T} ::= \mathsf{T} \to \mathsf{T} | \mathtt{Bool}$$

$(\to)$ は右結合的とする．

抽象の各引数には型注釈が施されているものとする．以下，λ 項に対する型付け規則について考える．[2]

- 抽象について．抽象 λ 抽象の引数の型が分かれば，関数の結果の型はその本体の型となる．これは次の規則で表わせる:

$$\frac{x : T_1 \vdash t_2 : T_2}{\vdash \lambda x : T_1 . t_2 : T_1 \to T_2} \text{ (T-ABS)}$$

  - 抽象は入れ子となるから，仮定は複数個となりうる．そのため，次に定義される型付け文脈 (型環境) のもとでの型導出を定める必要がある．

[2] このように，項の型注釈により型検査器を誘導する言語は明示的に型付けされた言語と呼ばれる．対して，明示的に型注釈が施されていない項に対して，この情報を推論 (i.e. 再構築) させる言語は暗黙に型付けされた言語と呼ばれる．

Def(型付け文脈)

型付け文脈 Γ は変数とその型の列である．空の文脈は $\varnothing$ と書かれ，(仮定が空であるとして) 省略される．Γ で束縛される変数の集合を $dom\ (\Gamma)$ と書く．
また，文脈と型付け関係を取り，新たな文脈を返す演算子 $(,)$ が定義されている．$(,)$ は Γ の右 に新しい束縛を加えて拡張する．

Rem.
新しい束縛と既に Γ に現れている束縛の衝突を防ぐため，名前 x は Γ で束縛されている変数とは異なるように選ぶ．これは抽象に適宜 α 変換を施すことで達成できる．

以上を踏まえ，T-ABS は次のように修正される．また，変数の型付け規則も決まる:

$$\frac{\Gamma, x : T_1 \vdash t_2 : T_2}{\Gamma \vdash \lambda x : T_1.t_2 : T_1 \to T_2}\ (\text{T-ABS})$$

$$\frac{x : T \in \Gamma}{\Gamma \vdash x : T}\ (\text{T-VAR})$$

型付け関係

- 関数適用は，適用項と被適用項の自由変数の型を揃える必要があることに注意すれば，次のように定まる:

$$\frac{\Gamma \vdash \mathsf{t}_1 : \mathsf{T}_1 \to \mathsf{T}_2 \qquad \Gamma \vdash \mathsf{t}_2 : \mathsf{T}_1}{\Gamma \vdash \mathsf{t}_1\ \mathsf{t}_2 : \mathsf{T}_2}\ (\text{T-APP})$$

- 他の規則は変わらない.

以上の内容を，次ページに再度示す.

$\lambda_{\to}$ の型付け関係

図 9.1

構文

$t ::= x | \lambda x : T.t | t$(項の構文)

$v ::= \lambda x : T.t$(値の構文)

$T ::= T \to T$(型の構文)

$\Gamma ::= \varnothing | \Gamma, x : T$(文脈の構文)

評価

$$\frac{t_1 \to t_1'}{t_1\ t_2 \to t_1'\ t_2}\ \text{E-APP1}$$

$$\frac{t_2 \to t_2'}{v_1\ t_2 \to v_1\ t_2'}\ \text{E-APP2}$$

$$(\lambda x : T_{11}.t_{12})\ v_2 \to [x \mapsto v_2]\ t_{12}$$

型付け

$$\frac{x : T \in \Gamma}{\Gamma \vdash x : T}\ \text{(T-VAR)}$$

$$\frac{\Gamma, x : T_1 \vdash t_2 : T_2}{\Gamma \vdash \lambda x : T_1.t_2 : T_1 \to T_2}\ \text{(T-ABS)}$$

$$\frac{\Gamma \vdash t_1 : T_1 \to T_2 \qquad \Gamma \vdash t_2 : T_1}{\Gamma \vdash t_1\ t_2 : T_2}\ \text{(T-APP)}$$

算術式の場合同様に逆転補題が成り立つ．証明は，各仮定を満たす規則が一意的であることより従う．

Lem.9.3.1.(型付けの逆転補題)

- $\Gamma \vdash x : R \Longrightarrow x : R \in \Gamma$.
- $\Gamma \vdash \lambda x : T_1.t_2 : R \Longrightarrow \exists R_2.\Gamma, x : T_1 \vdash t_2 : R_2 \wedge R = T_1 \to R_2$.
- $\Gamma \vdash t_1\ t_2 : R \Longrightarrow \exists T_{11}.\Gamma \vdash t_1 : T_{11} \to R \wedge \Gamma \vdash t_2 : T_{11}$
- $\Gamma \vdash \mathtt{true} : R \Longrightarrow R = \mathtt{Bool}$
- $\Gamma \vdash \mathtt{false} : R \Longrightarrow R = \mathtt{Bool}$
- $\Gamma \vdash \mathtt{if}\ t_1\ \mathtt{then}\ t_2\ \mathtt{else}\ t_3 : R \Longrightarrow \Gamma \vdash t_1 : \mathtt{Bool} \wedge \Gamma \vdash t_2, t_3 : R$

演習 9.3.2.

演習.9.3.2
$\Gamma \vdash x\ x : T$ なる Γ, T は存在するか.

Proof.
存在しないことを示す．このような Γ，型 T が存在すると仮定し，これらを取る．逆転補題より，ある型 T_1' が存在し，$x : T_1' \to T_1 \wedge x : T_1'$ が成り立ち，この T_1' を取ることで $\Gamma = x : T_1' \to T, x : T_1'$ と書ける．ところが，Γ 中の x の束縛は一意的なので $T_1' = T_1' \to T_1$ が成り立つ．したがって，T_1' は無限の大きさを持てることになる．矛盾．ゆえにこのような Γ, T は存在しない． $\square$

型の一意性

Thm.9.3.3.(型の一意性)
型付け文脈 Γ が 1 つ与えられると，それに対して 1 つの項 t は 1 つの型を持つ．ただし，$FV(t) \subseteq dom(\Gamma)$．すなわち，ある項が型付け可能ならば，その型は一意に定まる．さらに，型付け関係を生成する推論規則からただ一つの型付け導出が構築できる．

証明.
型付け導出に関する帰納法による．$FV(t) \subseteq dom(\Gamma)$ なる項 t に対する $t : T$ の導出に関する帰納法で示す．任意の Γ に対し，$s : S$ が $t : T$ の部分導出で証明されるとき，S が一意的であることを仮定する．このとき，導出の最後で使われた型付け規則で分ける．t が基本型または条件文の場合は算術式同様なので，λ 項の場合について示す．

1. T-VAR のとき．逆転補題より $t : T \in \Gamma$ となり，仮定から t の型が一意的なることより従う．
2. T-ABS のとき，$t = \lambda x : T_1.t_2 : T$ なる $x : T_1, t_2 : T_2$ が存在し，$T : T_1 \to T_2$．仮定より，導出 $\Gamma, x : T_1 \vdash t_2 : T_2$ は一意的なので示される．
3. T-APP のとき．$t = t_1\ t_2$ なる項 t_1, t_2 で，$\Gamma \vdash t_1 : T_{11} \to T$，$\Gamma \vdash t_2 : T_{11}$ なるものが取れる．帰納法の仮定より，これらの型付け導出は一意的なので従う．

Lem.9.3.4.(標準形)

1. v が型 Bool の値ならば，v は true または false のどちらかである．
2. v が型 $T_1 \to T_2$ の標準形ならば，$v = \lambda x : T_1.t_2$ である．

Proof.
値は true, false および $\lambda x : T.t$ の形を持つ．

1. (1) について．true, false の場合は明らかである．また，$\lambda x : T.t : Bool$ の形にはなりえない．それは，逆転補題よりある R が存在し，$x : T_1 \vdash t : R \wedge Bool = T_1 \to R$ が成り立ち，基本型たる Bool が関数型となることはできないため．
2. (2)．後者の場合は明らか．前者の場合は，$T_1 \to T_2 = Bool$ が逆転補題から成り立ち，先程と同じ理由から成立しないことから従う．

□

Thm.9.3.5.(進行)

$\mathtt{t}$ を閉じた，正しく型付けされた項とする.(i.e.$\exists T, \vdash \mathtt{t} : \mathsf{T}$)．このとき $\mathtt{t}$ は値か，$\exists \mathtt{t}'.\mathtt{t} \to \mathtt{t}'$ を満たす．

Proof.
型付け導出に関する帰納法による．ブール値および条件文に対する型付け規則については，算術式と同様の理由により成立する．仮定より，$\mathtt{t}$ に変数は出現しえず，値である．ラムダ抽象は値なので，即座に成立する．

関数適用の場合を考える．$\mathtt{t} = \mathtt{t}_1\ \mathtt{t}_2$ で $\vdash \mathtt{t}_1 : \mathsf{T}_{11} \to \mathsf{T}_{12} \wedge \vdash \mathtt{t}_2 : \mathsf{T}_{11}$ とする．帰納法の仮定より，$\mathtt{t}_1$ は値か，1 ステップ評価が可能である．

- 1 ステップ評価が可能のとき．$\mathrm{E-APP1}$ が適用され，題意が従う．
- 値のとき．
  - $\mathtt{t}_2$ が 1 ステップ評価可能のとき．$\mathrm{E\text{-}APP2}$ より評価が 1 ステップ可能．
  - 値のとき．標準形補題により $\mathtt{t}_1$ は $\lambda \mathtt{x} : \mathsf{T}_{11}.\mathtt{t}_{12}$ の形であり，$\mathrm{E\text{-}APPABS}$ が適用される．

□

保存

評価が型を保存することを示そう．準備として，型付け文脈に対して水増しと交換を施しても導出される型付け判断式の結論は変わらないことを確認する．

Lem.9.3.6.(並べ替え)
$\Gamma \vdash t : T$ かつ Δ が Γ の並べ替えならば，$\Delta \vdash t : T$ であり，その導出の深さは仮定を Γ とした場合と変わらない．

Lem.9.3.7.
$\Delta \vdash t : T$ かつ $x \notin dom\ (\Gamma)$ ならば，$\Gamma, x : S \vdash t : T$ であり，後者と前者の導出の深さは等しい．

Proof.
証明は induction による (白板で行なう) □

代入補題

Lem.9.3.8.(代入の下での型の保存)
$\Gamma, x : S \vdash t : T$ かつ $\Gamma \vdash s : S$ ならば $\Gamma \vdash [x \mapsto s]\, t : T$.

Proof.
$\Gamma \vdash [x \mapsto s]\, t : T$ の導出の深さに関する帰納法による．最後に使われた規則で分ける．

- $\mathrm{T\text{-}VAR}$ のとき．$z : T \in (\Gamma, x : S)$ なる z により $t = z$ と書ける．$z = x$ の場合は代入結果は $s : S$ であり，仮定より即座に従う．$z \neq x$ の場合，結果は $z : T$ であり，これも即座に従う．
- $\mathrm{T\text{-}ABS}$ のとき．$t = \lambda y : T_2.t_1, T = T_2 \to T_1$ かつ $\Gamma, x : S, y : T_2 \vdash t_1 : T_1$ が成り立つ．$x \neq y \wedge y \notin FV(s)$ としてよい．仮定を並び替えると $\Gamma, y : T_2, x : S \vdash t_1 : T_1$．また $\Gamma \vdash s : S$ に弱化を施すと $\Gamma, y : T_2 \vdash s : S$．帰納法の仮定により $\Gamma, y : T_2 \vdash [x \mapsto s]\, t_1 : T_1$．ゆえに代入の定義および $\mathrm{T\text{-}ABS}$ より，$\Gamma \vdash \lambda y : T_2.\, [x \mapsto s]\, t_1 : T_2 \to T_1$.

□

代入補題

Proof.

- $\mathrm{T\text{-}APP}$ のとき．$\mathsf{t} = \mathsf{t}_1\ \mathsf{t}_2$ かつ $\Gamma, \mathsf{x} : \mathsf{S} \vdash \mathsf{t}_1 : \mathsf{T}_2 \to \mathsf{T}_1, \mathsf{t}_2 : \mathsf{T}_2$ かつ $\mathsf{T} = \mathsf{T}_1$．帰納法の仮定より $\Gamma \vdash [\mathsf{x} \mapsto \mathsf{s}]\,\mathsf{t}_1 : \mathsf{T}_2 \to \mathsf{T}_1$ かつ $\Gamma \vdash [\mathsf{x} \mapsto \mathsf{s}]\,\mathsf{t}_2 : \mathsf{T}_2$．$\mathrm{T\text{-}APP}$ より題意が従う．
- $\mathrm{T\text{-}TRUE|FALSE}$ のとき．前者のみ確認すればよい．このとき $\mathsf{t} = \mathtt{true}$ かつ $\mathsf{T} = \mathtt{Bool}$．$[\mathsf{x} \mapsto \mathsf{s}]\,\mathsf{t} = \mathtt{true}$ より，即座に成立する．
- $\mathrm{T\text{-}IF}$ のとき．$\mathsf{t} = \mathtt{if}\ \mathsf{t}_1\ \mathtt{then}\ \mathsf{t}_2\ \mathtt{else}\ \mathsf{t}_3$ かつ $\Gamma, \mathsf{x} : \mathsf{S} \vdash \mathsf{t}_1 : \mathtt{Bool}$ かつ $\Gamma, \mathsf{x} : \mathsf{S} \vdash \mathsf{t}_2 : \mathsf{T}$ かつ $\Gamma, \mathsf{x} : \mathsf{S} \vdash \mathsf{t}_3 : \mathsf{T}$．帰納法の仮定より $\Gamma \vdash [\mathsf{x} \mapsto \mathsf{s}]\,\mathsf{t}_1 : \mathtt{Bool}$，$\Gamma \vdash [\mathsf{x} \mapsto \mathsf{s}]\,\mathsf{t}_2 : \mathsf{T}$ かつ $\Gamma \vdash [\mathsf{x} \mapsto \mathsf{s}]\,\mathsf{t}_3 : \mathsf{T}$．$\mathrm{T\text{-}IF}$ より従う．

□

型保存

Thm.9.3.9.(型保存)
$\Gamma \vdash t : T$ かつ $t \to t'$ ならば $\Gamma \vdash t' : T$.

Proof.
$\Gamma \vdash t : T$ の導出に関する帰納法による．いま，$t : T$ の任意の直接の部分導出に現れる $s : S$ に対して，$\exists s'.s \to s'$ ならば $s' : S$ を仮定し，最後に使われた規則で場合を分ける．

- T-VAR．左辺に変数を持つ規則は存在しないので，起きえない．
- T-ABS．結論は値なので起きえない．
- T-APP．その評価に使われた規則で分ける．
  - E-APP1(関数部の評価) のとき．$t_1 \to t_1'$ なる t_1' が取れる．仮定より $t_1' : T_1$．したがって T-APP より $t' = t_1'\ t_2 : T$ となる．
  - E-APP2 のとき．$t_2 \to t_2'$ なる t_2' が取れる．仮定より $t_2' : T$．したがって E-APP1 同様にして $t' = v_1\ t_2 : T$.
  - E-APPABS のとき．$t_1 = \lambda x : T_{11}.t_{12}, t_2 = v_2, t' = [x \mapsto v_2]\, t_{12}$．逆転補題より $\lambda x : T_{11}.t_{12}$ の型付け導出を分解し，$\Gamma, x : T_{11} \vdash t_{12} : T_{12}$ を得る．代入補題より従う．
  - 他の場合はブール式の場合に等しい．

□

演習 9.3.10.
主部展開は STLC の”関数的な部分”では成り立つだろうか．i.e.t が条件式を含まないと仮定したとき，$t \to t'$ かつ $\Gamma \vdash t' : T$ ならば $\Gamma \vdash t : T$ だろうか．

解答．．
成り立つ．条件式を含まない t を任意に取り，$t \to t' \wedge \Gamma \vdash t' : T$ を仮定する．subterm が正規形か否かで場合を分ける．

1.

□

型と命題，項とその構成的証明を対応づけるのが Curry-Howard 対応であった．以下では，STLC の型付け規則が直観主義的な implicational logic に対応することを確認する．型付け規則との対応を明白にするために，今回はシークエント計算を用いる．

$\Gamma, \Delta, \Pi, \Sigma$ を論理式の集合，A, B を論理式とすると，含意に関する推論規則は次の通りだった:

$$\frac{\Gamma, A, \Delta \vdash \Pi, B, \Sigma}{\Gamma, \Delta \vdash \Pi, A \Rightarrow B, \Sigma} (\Rightarrow \mathrm{R})$$

$$\frac{\Gamma \vdash A, \Delta \qquad \Sigma, B \vdash \Pi}{\Gamma, \Sigma, A \Rightarrow B \vdash \Delta, \Pi} (\Rightarrow \mathrm{L})$$

これが，型付け規則 T-ABS, T-APP と対応することを確認する．
$\Delta = \Sigma = \Pi = \varnothing$ と仮定してよい．$(\Rightarrow \mathrm{R})$ と T-ABS の対応は，A, B をそれぞれ変数 $\mathsf{x} : \mathsf{T}$，項 $\mathsf{t}_2 : \mathsf{T}_2$ に読みかえれば分かる．

T-APP について．$\Delta = \varnothing, \Sigma = \Gamma \setminus B, \Pi = \{B\}$ とすると，$(\Rightarrow \mathrm{L})$ は，自然演繹の以下の証明図に等しい:

$$\dfrac{\dfrac{\Gamma}{A} \qquad \dfrac{}{A \Rightarrow B}}{\dfrac{B}{B}\,(\Rightarrow \mathrm{E})} \qquad \dfrac{}{\Gamma}$$

重複した論理式の集合を除き (これは型環境の構成からいつでも可能)，$A = \mathsf{t}_2 : \mathsf{T}_1, \mathsf{A} \Rightarrow \mathsf{B} = \mathsf{t}_1 : \mathsf{T}_1 \to \mathsf{T}_2$ とすれば，$B = \mathsf{t}_1\ \mathsf{t}_2 : \mathsf{T}_2$ から従う．

最後に，β 簡約が cut 除去に対応することを確かめる．

実際，導出木の根に β 簡約を適用すると，cut 則の適用を含む任意の導出木

$$\dfrac{\dfrac{\vdots}{\Gamma \vdash \mathsf{v}_1 : \mathsf{T}_1} \qquad \dfrac{\dfrac{\vdots}{\Gamma, \mathsf{x} : \mathsf{T}_1 \vdash \mathsf{t}_{12} : \mathsf{T}_2}}{\Gamma \vdash \lambda \mathsf{x} : \mathsf{T}_1 . \mathsf{t}_{12} : \mathsf{T}_1 \to \mathsf{T}_2}}{\Gamma \vdash (\lambda \mathsf{x} : \mathsf{T}_1 . \mathsf{t}_{12}) \mathsf{v}_1 : \mathsf{T}_2} \, (\mathrm{cut})$$

に対して，cut を含まない導出木

$$\dfrac{\vdots}{\Gamma \vdash [\mathsf{x} \mapsto \mathsf{v}_1] \, \mathsf{t}_{12} : \mathsf{T}_2}$$

を構成できる．以上から，確かに STLC の型付け規則が LJ の $(\Rightarrow)$ のみを含む subset と一対一対応することが示される．

References

- Semantics of Programming Languages: Structures and Techniques, C.Gunter, MIT Press, 1992.
- Proofs and Types, J.Y. Girard, Y. Lafont and P.Taylor, Cambridge Tracts in Theoretical Computer Science, 1987.