コンピュータサイエンス入門
～アルゴリズムとプログラミング言語～
2020 年度 Q2 解答例と解説 (再配布しないで下さい)

2020 年 8 月 16 日

問 10.1(p. 137–138) p. 137 に以下のように関数の定義がある．

> f が A から B への関数であるのは，任意の $x \in A$ について $(x, y) \in f$ となる $y \in B$ があり，任意の $x \in A, y \in B, z \in B$ について $(x, y) \in f$ かつ $(x, z) \in f$ なら $y = z$ となるときである．

つまり，論理式で書くと，f が A から B への関数であるのは

$$(\forall x \in A. \exists y \in B.\ (x, y) \in f)\ \wedge\ (\forall x \in A, y, z \in B.\ (x, y) \in f \wedge (x, z) \in f \Rightarrow y = z)$$

と同値である．R が関数であることから，

1. 任意の x について $(x, z) \in R$ となる $z \in A$ があり，
2. 任意の $x, z, z' \in A$ について $(x, z) \in R$ かつ $(x, z') \in R$ なら $z = z'$ となる．

同様に，S が関数であることから，

3. 任意の z について $(z, y) \in S$ となる y があり，
4. 任意の $z, y, y' \in A$ について $(z, y) \in S$ かつ $(z, y') \in S$ なら $z = z'$ となる．

1 と 3 より，任意の x について $(x, y) \in RS$ となる y がある．また，2 と 4 より任意の x, y, y' について $(x, y) \in RS$ かつ $(x, y') \in RS$ なら $y = y'$ となる．したがって RS は関数である．

以下のように式に名前を付ける．

$$R^* = \bigcup \{R^n \mid n \geq 0\} \tag{1}$$

R の反射的推移的閉包とは (1)R を含み，(2) 反射的かつ (3) 推移的な関係の中で (4) 最小のものである．R^* が R の反射的推移的閉包であることを (1)–(4) を示すことで示す．

1. R^* は R を含むこと，すなわち $R^* \supseteq R$ を示す：
   $R^* = \bigcup \{R^n \mid n \geq 0\} \supseteq R^k \supseteq R^1 = R$
2. R^* が反射的であること，すなわち任意の $x \in A$ に対して $(x, x) \in R^*$ であることを示す：
   $R^* \supseteq R^0 = \{(x, x) \mid x \in A\}$ より明らか．
3. R^* が推移性を満たすことを示す：
   $(x, y), (y, z) \in R^*$ とする．R^* の定義よりある k_1, k_2 が存在して，$(x, y) \in R^{k_1}$ かつ $(y, z) \in R^{k_2}$ である．ここで，関係には結合側が成り立つことから任意の非負整数 k_1 と k_2 に対して $R^{k_1+k_2} = R^{k_1} R^{k_2}$ である．したがって，$(x, z) \in R^{k_1+k_2} \subseteq R^*$ である．
4. R' を R を含み反射的かつ推移的な任意の関係とし $R^* \subseteq R'$ を示す：
   任意の非負整数 n (≥ 0) に対して $R^n \subseteq R'$ を示せば十分である．n に関する帰納法で示す．

$n = 0$ の場合を考える．R' は反射的なので $R^0 \subseteq R'$ である．

$n > 0$ の場合を考える．帰納法の仮定より $R^{n-1} \subseteq R'$ である．また R' が反射的な R を含む反射的かつ推移的な関係であることから $R'R = R'$ である．よって $R^n = R^{n-1}R \subseteq R'R = R'$ である．したがって $R^n \subseteq R'$ である．

練習問題 問 11.1(p. 141) 次のうちから正しいものをすべて選べ．

1. $C = A \cup B$ のとき，$C \supseteq A$ である．
2. $C = A \cup B$ のとき，$C \ni A$ である．

練習問題 問 11.1(p. 141) 2 つの集合 $A = \{1, 2, 3\}$, $B = \{4, 5\}$ に対して，AB を求めよ．

解答例 $AB = \{14, 15, 24, 25, 34, 35\}$

問 11.1(p. 141) (i) $\emptyset = \mathcal{R}^0 \subseteq \bigcup\{\mathcal{R}^n \mid n \geq 0\} = \mathcal{R}$.

(ii) $\{\epsilon\} \in \mathcal{R}^1 \subseteq \bigcup\{\mathcal{R}^n \mid n \geq 0\} = \mathcal{R}$.

(iii) すべての $a \in \Sigma$ について，$\{a\} \in \{\{b\} \mid b \in \Sigma\} \subseteq \mathcal{R}^1 \subseteq \bigcup\{\mathcal{R}^n \mid n \geq 0\} = \mathcal{R}$.

(iv) $R \in \mathcal{R} = \{\mathcal{R}^n \mid n \geq 0\}$ であることから，ある k が存在して $R \in \mathcal{R}^k$ である．よって，$R^* \in \{S^* \mid S \in \mathcal{R}^k\} \subseteq \mathcal{R}^{k+1} \subseteq \{\mathcal{R}^n \mid n \geq 0\} = \mathcal{R}$ である．

(v) $R_1, R_2 \in \mathcal{R}$ であることから，ある k が存在して $R_1, R_2 \in \mathcal{R}^k$ である．よって，

$$R_1 R_2 \in \{S_1 S_2 \mid S_1 \in \mathcal{R}^k, S_2 \in \mathcal{R}^k\} \subseteq \mathcal{R}^{k+1} \subseteq \mathcal{R} \tag{2}$$

$$R_1 \cup R_2 \in \{S_1 \cup S_2 \mid S_1 \in \mathcal{R}^k, S_2 \in \mathcal{R}^k\} \subseteq \mathcal{R}^{k+1} \subseteq \mathcal{R} \tag{3}$$

である．

ある集合 A が (i) から (v) までの性質：

(i) $\emptyset \in A$

(ii) $\{\epsilon\} \in A$

(iii) すべての $a \in \Sigma$ について $\{a\} \in A$

(iv) $R \in A$ なら $R^* \in A$ である．

(v) $R_1, R_2 \in A$ なら $R_1 R_2 \in A$ かつ $R_1 \cup R_2 \in A$ である．

を満たすものとする．

任意の n について $\mathcal{R}^n \subseteq A$ であることを，任意の $S \in \mathcal{R}^n$ が $S \in A$ であることを n に関する帰納法により示す．

$n = 0$ の場合．前提が空であり成り立つ．

$n = k + 1$ の場合．帰納法の仮定より，任意の $S \in \mathcal{R}^k$ が $S \in A$ である．これと上の性質 (i) から (v) より漸化式の右辺の和を構成するどの集合の要素でも A に属することがいえる．よって，任意の $S \in \mathcal{R}^{k+1}$ が $S \in A$ である．

以上より，任意の n について $\mathcal{R}^n \subseteq A$ であることが示された．

（任意の n について $\mathcal{R}^n$ が離散集合であることから，）(i) から (v) までの性質を満たす任意の集合 A と任意の n について，$\mathcal{R}^n \subseteq A$ であり，したがって，$\mathcal{R} = \bigcup\{\mathcal{R}^n \mid n \geq 0\} \subseteq A$ である．この A の任意性により $\mathcal{R}$ は (i) から (v) を満たす最小の集合である．

練習問題 問 11.2 以下は高校の数学 I[1, pp. 55–] の問である．

問 62 $A = \{a, b, c\}$ のとき，A の部分集合をすべて書け．

全体集合 U の部分集合 A について，A に属さない U の要素全体の集合を，A の**補集合**といい，$\overline{A}$ で表す．すなわち，

$$\overline{A} = \{x \mid x \in U, x \notin A\}$$

問 63 $U = \{x \mid x$ は 10 以下の自然数 $\}$ を全体集合とするとき，

$$A = \{x \mid x \in U, x \text{ は } 2 \text{ の倍数 }\}$$
$$B = \{x \mid x \in U, x \text{ は } 3 \text{ の倍数 }\}$$

に対して，次の集合を，要素を書き並べて表せ.

(1) $\overline{A}$　　(2) $A \cap \overline{B}$　　(3) $\overline{A} \cup B$

問 64 $\overline{A \cap B} = \overline{A} \cup \overline{A}$ が成り立つことを示せ.

問 65 12 以下の自然数の集合を全体集合 U とする．U の部分集合

$$A = \{x \mid x \in U, x \text{ は } 2 \text{ の倍数 }\}$$
$$B = \{x \mid x \in U, x \text{ は } 3 \text{ の倍数 }\}$$

について，要素を書き並べて，ド・モルガンの法則が成り立つことを確かめよ.

問 11.2(p. 141) 性質 (i) の証明のヒント：数学的帰納法を用いる．性質 (iv) の証明のヒント：全体集合を Σ^* として補集合を考えて，任意の集合 X $(\subset \Sigma^*)$ がその補集合の補集合 $\Sigma^* \backslash (\Sigma^* \backslash X)$ と等しいこととド・モルガンの法則を用いると良い.

性質 (i) の証明．R が正規言語であることから，p. 140 の性質 (ii) と (v) および $\cdot^n$ の定義を用いた n に関する帰納法により，任意の n について R^n は正規言語である.

性質 (ii) の証明．R が正規言語であることから，p. 140 の性質 (iv) より R^* は正規言語である．よって，p. 140 の性質 (v) より，RR^* すなわち R^+ は正規言語である.

性質 (iv) の証明．R_1, R_2 が正規言語であることから，性質 (iii) より $\Sigma^* \backslash R_1$ と $\Sigma^* \backslash R_2$ が正規言語であり，p. 140 の性質 (v) より $(\Sigma^* \backslash R_1) \cup (\Sigma^* \backslash R_2)$ は正規言語である．よって

$$\begin{aligned} R_1 \cap R_2 &= \Sigma^* \backslash (\Sigma^* \backslash (R_1 \cap R_2)) && \because R_1 \cap R_2 \subseteq \Sigma^* \\ &= \Sigma^* \backslash ((\Sigma^* \backslash R_1) \cup (\Sigma^* \backslash R_2)) && \because \text{ド・モルガンの法則} \end{aligned}$$

も性質 (iii) より正規言語である.

問 11.3(p. 141) $X = \bigcup \{R^n S \mid n \geq 0\}$ が解であり，$\epsilon \notin R$ より唯一の解である．p. 140 の性質 (v) を用いた帰納法により X は正規言語であることが示される.

なお，$\epsilon \in R$ である場合は，任意の集合 X が $X = S \cup RX$ を満たす.

練習問題 例 11.2(p. 142) -1234, 3.14, $1.09E20$ が $[\![number]\!]$ の要素であることを示せ.

解答例 $-1234 \in [\![number]\!]$ であるときのみを示す．他も同様である.

$$\begin{aligned} [\![number]\!] &= [\![int]\!] \\ &= [\![sign\ \ digit^+]\!] \\ &= [\![sign]\!]\ [\![digit^+]\!] \\ &= [\![+ \mid\ - \mid \epsilon]\!]\ [\![digit]\!]^+ \\ &= ([\![+]\!] \cup [\![-]\!] \cup [\![\epsilon]\!])\ \left(\bigcup \{[\![digit]\!]^n \mid n \geq 1\}\right) \\ &\supset [\![-]\!]\ [\![digit]\!]^4 \\ &= \{-\}\ [\![digit]\!]\ [\![digit]\!]^3 \\ &= \cdots \\ &= \{-\}\ [\![digit]\!]\ [\![digit]\!]\ [\![digit]\!]\ [\![digit]\!]\ [\![digit]\!]^0 \\ &= \{-\}\ [\![digit]\!]\ [\![digit]\!]\ [\![digit]\!]\ [\![digit]\!]\ \{\epsilon\} \\ &\supset \{-\}\ [\![1]\!]\ [\![2]\!]\ [\![3]\!]\ [\![4]\!]\ \{\epsilon\} \\ &= \{-1234\} \\ &\ni -1234 \end{aligned}$$

問 11.4(p. 146) w の長さに関する帰納法を用いる.

w の長さが 0 の場合：$\hat{\delta}$ の定義から明らか.

w の長さが 1 より大きい場合：w を aw' とする．

$$\begin{aligned}
\hat{\delta}(aw'v, q) &= \hat{\delta}(w'v, \delta(a, q)) && \because \hat{\delta}\text{の定義} \\
&= \hat{\delta}(v, \hat{\delta}(w', \delta(a, q))) && \because \text{帰納法の仮定} \\
&= \hat{\delta}(v, \hat{\delta}(aw', q)) && \because \hat{\delta}\text{の定義}
\end{aligned}$$

以上より題意は示された．

問 11.5(p. 146) ヒント：各状態において各終端記号による遷移を考えると良い．
問 11.6(p. 147) **問 11.5** で作成した有限状態機械に対応する相互再帰関数を作成する．

$$\begin{aligned}
f_1(x :: L) &= f_2(L) && (x \in \{0,1,2,3,4,5,6,7,8,9\}) \\
f_1(x :: L) &= f_3(L) && (x \in \{+,-\}) \\
f_1(x :: L) &= f_8(L) && (x \in \{'.', 'E'\}) \\
f_1(nil) &= Reject \\
f_2(x :: L) &= f_2(L) && (x \in \{0,1,2,3,4,5,6,7,8,9\}) \\
f_2(x :: L) &= f_8(L) && (x \in \{+,-\}) \\
f_2('.' :: L) &= f_4(L) \\
f_2('E' :: L) &= f_5(L) \\
f_2(nil) &= Accept \\
f_3(x :: L) &= f_2(L) && (x \in \{0,1,2,3,4,5,6,7,8,9\}) \\
f_3(x :: L) &= f_8(L) && (x \in \{'.', +, -, 'E'\}) \\
f_3(nil) &= Reject \\
&\vdots
\end{aligned}$$

練習問題 問 11.7(pp. 150–151) BNF 文法

$$K ::= () \mid (K) \mid K\ K$$

で生成される構文木を OCaml のデータ型で表現せよ．
解答例 BNF 記法は「文脈自由文法を簡潔に定義する記法」(p. 144) である．上記の BNF 文法は文脈自由文法 $G_1 = (T_1, N_1, K, P_1)$ を定義している．ここで，

$$\begin{aligned}
N_1 &= \{K\} \\
T_1 &= \{(,)\} \\
P_1 &= \{K \to (), K \to (K), K \to K\ K)\}
\end{aligned}$$

である．「各非終端記号 N をデータ型とみなし，N の各生成規則に対してデータ構成子を定義する」(p.150) ということにしたがって，非終端記号 K をデータ型とみなし，生成規則 $K \to ()$ に対してデータ構成子 `OneParen`，生成規則 $K \to (K)$ に対してデータ構成子 `Paren`，生成規則 $K \to K\ K$ に対してデータ構成子 `Apply` をそれぞれ定義する．データ `OneParen` は，K をルートノードとし，“(”-木と “)”-木を子とする K-木を表す．データ `Paren(t)` は，K をルートノードとし，“(”-木，`t` が表す K-木，および “)”-木を子とする K-木を表す．データ `Apply(t1,t2)` は，K をルートノードとし，`t1` が表す K-木と `t2` が表す K-木を子とする K-木を表す．

以上より求めるデータ型 `k` は以下の通りである．

```
type k = OneParen | Paren of k | Apply of k * k
```

(※ OCaml ではデータ型は英小文字で開始する必要がある．)
練習問題 問 11.7(pp. 150–151) **問 11.7** の BNF 文法における非終端記号をすべて記せ．
解答例 *PROGRAM* と E の 2 つ．なお，タイプライターフォントで書かれたものはコードであることを明示しているものと思われる．

練習問題 問 11.7(pp. 150–151) **問 11.7** の BNF 文法における *id* と *int* をそれぞれ正規表現を用いて定義せよ．また，*id* と *int* はそれぞれ終端記号と非終端記号のいずれであるか．
解答例 *int* には，p. 142 **例 11.2** の定義を流用する．

$$
\begin{aligned}
\mathit{sign} &= \texttt{+} \mid \texttt{-} \mid \epsilon \\
\mathit{digit} &= \texttt{0} \mid \texttt{1} \mid \texttt{2} \mid \texttt{3} \mid \texttt{4} \mid \texttt{5} \mid \texttt{6} \mid \texttt{7} \mid \texttt{8} \mid \texttt{9} \\
\mathit{int} &= \mathit{sign}\ \mathit{digit}^{+}
\end{aligned}
$$

id は先頭が英小文字，2 文字目から英数字および_である 1 文字以上の文字列とする．

$$
\begin{aligned}
\mathit{upper} &= \texttt{A} \mid \cdots \mid \texttt{Z} \\
\mathit{lower} &= \texttt{a} \mid \cdots \mid \texttt{z} \\
\mathit{alphanum} &= \mathit{lower} \mid \mathit{upper} \mid \mathit{digit} \mid \texttt{\_} \\
\mathit{id} &= \mathit{lower}\ \mathit{alphanum}^{*}
\end{aligned}
$$

p. 143 の終端記号の定義では，「通常のプログラミング言語では T の各要素は正規表現で定義される集合を表す」とあるので *id* と *int* は両方とも終端記号である．

問 11.7(pp. 150–151) 解答は別途示す．
2020 年度に受講者がはまった点：

- Windows で環境構築が難しい → VirtualBox + Ubuntu を用いると良い
- OCaml のインストール方法が分からない．[2] のサポートページに記載の方法のうちバイナリパッケージをとってきてインストールする方法に失敗した．→ 公式ページの情報 `https://ocaml.org/docs/install.html` を参考にすると良い．VirtualBox + Ubuntu では apt でインストールのが簡単．
- OCaml インタプリタで日本語が文字化けをおこす．[2] のサポートページに記載の方法では文字化けが直らない． → ホームディレクトリに以下を記述した .ocamlinit を置く．

```
let print_non_escaped_string ppf = Format.fprintf ppf "\"%s\"";;
#install_printer print_non_escaped_string;;
```

- ファイルマネージャ Nautilus からダブルクリックで `*.ml` ファイルを開けない → この方法で開く必要は無い（この方法で開きたい人は「ファイルの関連付け」を調べること）
- コマンド ocaml でコンパイルできない → コマンド ocamlc でコンパイルすること
- GNU Make の使い方が分からない → [3] の 1 章だけでいいので読むこと．他の節は必要に応じて読むので良い．
- 最新の OCamlMakefile は `http://mmottl.github.io/ocaml-makefile/` から手に入る（2020/08/12 現在）
- 与えられた BNF 文法にしたがったプログラムを評価する方法が分からない → この問題では評価器を作る必要は無い
- ocamllex や ocamlyacc がよく分からない → まずはマニュアルを要チェック `https://ocaml.jp/archive/ocaml-manual-3.06-ja/manual026.html`
- 英単語：
  constructor: コンストラクタ，構成子; bind: 束縛; unbound: 束縛されていない
- LaTeX の問題．パッケージのインストール方法が分からない → tlmgr[*1]を使うのが簡単である．まず以下を理解すること．
  - パッケージ管理ツール
  - TeX Live
  - ヘルプの使い方 tlmgr –help, man tlmgr

  次に，以下のコマンドを理解した上で実行すること

[*1] `https://texwiki.texjp.org/?tlmgr`

```
sudo tlmgr update --self --all
sudo tlmgr install listings
```

ただし，tlmgr を初めて使うときは以下のコマンドを実行する必要がある[*2]．

```
sudo apt install xzdec
tlmgr init-usertree
```

- apt コマンドを使っていて「E: ロック /var/lib/dpkg/lock-frontend が取得できませんでした - open (11: リソースが一時的に利用できません)」というエラーが出る → https://qiita.com/tetsuo_jp/items/9d4bbf415f3ec900b894

「Minimal でも十分プログラミング可能であるが，ML に興味のある読者は，巻末にいくつかあげる ML に関する参考文献を参照し，ML でプログラミングしてみることを勧める」とある．参考文献は [4, 5, 6, 7] である．しかし，本演習では Minimal や Standard ML ではなく OCaml でプログラミングしてみることを勧める．さらに参照する参考文献として [2] を勧める．

なお，文献 [4, 5] はオンラインで PDF を閲覧可能である．[4, 5, 6] は ML のテキストであり，[7] は OCaml の前身の Caml を使って書かれたアルゴリズムの教科書である．

問 12.1(p. 155) p. 154 の与えられたラムダ式の文法は以下の通りである：

$$\begin{aligned} M ::= {} & c \mid x \mid \lambda x.M \mid M\ M \mid (M, M) \mid M[1] \mid M[2] \\ & \mid 1(M) \mid 2(M) \mid (\textbf{case}\ M\ \textbf{of}\ 1(x) \Rightarrow M, 2(x) \Rightarrow M) \\ & \mid \textbf{let}\ x = M\ \textbf{in}\ M \mid \textbf{fix}(M) \end{aligned}$$

まずは曖昧なラムダ式を考えてみよう：

- $\lambda x.M\ M$ は $(\lambda x.M)\ M$ と $\lambda x.(M\ M)$
- $M\ M[1]$ は $M\ (M[1])$ と $(M\ M)[1]$
- $M\ M[2]$ は $M\ (M[2])$ と $(M\ M)[2]$

にそれぞれ解釈できる．

直前の約束 (i)–(iii) が無くてもそれぞれの約束に対応する構文構造が一意に定まるようにすればよい．具体的には，$\lambda x.M$, $M\ M$, $M[1]$, $M[2]$ の周りに括弧を付けると良い．したがって，求める文法は以下の通りである：

$$\begin{aligned} M ::= {} & c \mid x \mid (\lambda x.M) \mid (M\ M) \mid (M, M) \mid (M[1]) \mid (M[2]) \\ & \mid 1(M) \mid 2(M) \mid (\textbf{case}\ M\ \textbf{of}\ 1(x) \Rightarrow M, 2(x) \Rightarrow M) \\ & \mid \textbf{let}\ x = M\ \textbf{in}\ M \mid \textbf{fix}(M) \end{aligned}$$

問 12.2(p. 157) 束縛変数を含まない他の 7 つの場合は以下の赤字の部分である：

$$\begin{aligned} M ::= {} & \color{red}{c} \mid x \mid \lambda x.M \mid M\ M \mid \color{red}{(M, M)} \mid \color{red}{M[1]} \mid \color{red}{M[2]} \\ & \mid \color{red}{1(M)} \mid \color{red}{2(M)} \mid (\textbf{case}\ M\ \textbf{of}\ 1(x) \Rightarrow M, 2(x) \Rightarrow M) \\ & \mid \textbf{let}\ x = M\ \textbf{in}\ M \mid \color{red}{\textbf{fix}(M)} \end{aligned}$$

[*2] https://texwiki.texjp.org/?Linux%2FLinux%20Mint

これらの場合の代入の定義は以下の通りである：

$$
\begin{aligned}
[N/x]c &= c \\
[N/x](M_1, M_2) &= ([N/x]M_1, [N/x]M_2) \\
[N/x](M[1]) &= ([N/x]M)[1] \\
[N/x](M[2]) &= ([N/x]M)[2] \\
[N/x](1(M)) &= 1([N/x]M) \\
[N/x](2(M)) &= 2([N/x]M) \\
[N/x](\mathbf{fix}(M)) &= \mathbf{fix}([N/x](M))
\end{aligned}
$$

問 12.3(p. 157) 場合分け構文と値の束縛構文は以下の赤字の部分である：

$$
\begin{aligned}
M ::= {} & c \mid x \mid \lambda x.M \mid M\ M \mid (M, M) \mid M[1] \mid M[2] \\
& \mid 1(M) \mid 2(M) \mid \textbf{(case } M \textbf{ of } 1(x) \Rightarrow M, 2(x) \Rightarrow M) \\
& \mid \textbf{let } x = M \textbf{ in } M \mid \mathbf{fix}(M)
\end{aligned}
$$

場合分け構文に対する，最も一般的な場合の規則のみを示す:

$$
\begin{aligned}
&[N/x](\textbf{case } M \textbf{ of } 1(y_1) \Rightarrow M_1, 2(y_2) \Rightarrow M_2) = \\
&\qquad \textbf{case } [N/x]M \textbf{ of } 1(z_1) \Rightarrow [N/x]([z_1/y_1]M_1), 2(z_2) \Rightarrow [N/x]([z_2/y_2]M_2) \\
&\qquad\qquad (x \neq y_i, z_i \neq x, y_i \in FV(N), z_i \notin FV(M_1) \cup FV(M_2) \cup FV(N))
\end{aligned}
$$

値の束縛構文に対する規則は以下の通りである：

$$
[N/x](\textbf{let } y = M_1 \textbf{ in } M_2) = \begin{cases}
\textbf{let } y = [N/x]M_1 \textbf{ in } M_2 & (x = y) \\
\textbf{let } y = [N/x]M_1 \textbf{ in } [N/x]M_2 & (x \neq y, y \notin FV(N)) \\
\textbf{let } z = [N/x]M_1 \textbf{ in } [N/x]([z/y]M_2) & (x \neq y, z \neq x, y \in FV(N) \text{ かつ} \\
 & \quad z \notin FV(M_2), z \notin FV(N))
\end{cases}
$$

問 12.4(p. 158) 場合分け構文に対する α 同値公理は次の通りである：

$$
\begin{aligned}
&\textbf{case } M \textbf{ of } 1(x_1) \Rightarrow M_1, 2(x_2) \Rightarrow M_2 \quad = \quad \textbf{case } M \textbf{ of } 1(y_1) \Rightarrow [y_1/x_1]M_1, 2(x_2) \Rightarrow M_2 \qquad (y_1 \notin FV(M_1)) \\
&\textbf{case } M \textbf{ of } 1(x_1) \Rightarrow M_1, 2(x_2) \Rightarrow M_2 \quad = \quad \textbf{case } M \textbf{ of } 1(x_1) \Rightarrow M_1, 2(y_2) \Rightarrow [y_2/x_2]M_2 \qquad (y_2 \notin FV(M_2))
\end{aligned}
$$

値の束縛構文に対する α 同値公理は次の通りである：

$$
\textbf{let } x = M_1 \textbf{ in } M_2 \quad = \quad \textbf{let } y = M_1 \textbf{ in } [y/x]M_2 \qquad (y \notin FV(M_1) \cup FV(M_2))
$$

問 12.5(p. 158) 先に定義したラムダ式の代入では $[\lambda x.x\ y/z]\lambda w.w\ z\ z$ のような代入を行うと $\lambda w.w\ (\lambda x.x\ y)\ (\lambda x.x\ y)$ となる．この結果のラムダ式は同じ束縛変数 x が 2 カ所で現れるので仮定 12.1 を満たさない．

関数適用に対する規則を以下のように変更する：

$$
\begin{aligned}
&[N/x](M_1\ M_2) = [N/x]M_1\ [[y_n/x_n] \cdots [y_1/x_1]N/x]M_2 \\
&\qquad (\forall i.\ y_i \notin FV([N/x]M_1) \cup BV([N/x]M_1) \cup FV(N) \cup BV(N) \cup FV(M_2) \cup BV(M_2))
\end{aligned}
$$

ここで y_i はフレッシュであることを意図している．ラムダ式の構造に関する帰納法によって仮定 12.1 を満たすことを示せる．

問 12.6(p. 159) 命題 p「a^b が有理数であるような無理数[*3]a, b が存在する」の否定 $\neg p$ は「任意の無理数 a, b に対して，a^b は有理数ではない」である．論理式で書くと $p = \exists a, b \in \mathbb{R}/\mathbb{Q}.a^b \in \mathbb{Q}$ の否定 $\neg p$ は $\forall a, b \in \mathbb{R}/\mathbb{Q}.a^b \notin \mathbb{Q}$ である．

背理法で p を示す．$\neg p$ であることを仮定する．すなわち，無理数 a, b が与えられたとき a^b は無理数と仮定する．$\sqrt{2}$ は無理数であるので，仮定より $\sqrt{2}^{\sqrt{2}}$ は無理数である．しかし，$a = \sqrt{2}^{\sqrt{2}}, b = \sqrt{2}$ と仮定すると a^b は，2 であり無理数ではない．よって矛盾．背理法により題意は示された．

問 12.7(p. 162) $\mathcal{N} \vdash \Delta \rhd A$ なら $\Delta \models A$ であることを，命題論理式 A の構造に関する帰納法で示す．

$A = a$ の場合．$\mathcal{N} \vdash \Delta \rhd A$ である．つまり，$\Delta \rhd a$ が $\mathcal{N}$ によって証明可能である． $\mathcal{N}$ の規則（図 12.2）のうち，$\Delta \rhd a$ に合致する結論を持ち得るのは規則 (Ax) のみである．したがって，$a \in Ax$ である．$a \in \Delta$ より，任意の η について $[\![a]\!]\eta = \mathit{true}$ である．すなわち，$\Delta \models a$ である．

$A = \alpha$ の場合．規則 (taut) によって $\alpha \in \Delta$ である．したがって，$\Delta \models \alpha$ である．

$A = X \supset Y$ の場合．規則 ($\supset$:I) より $\Delta \cup X \rhd Y$ である．これと帰納法の仮定より $\Delta \cup X \models Y$，すなわち

$$(\forall Z \in \Delta \cup X.\ [\![Z]\!]\eta = \mathit{true}) \text{ ならば } [\![Y]\!]\eta = \mathit{true}$$

である．したがって，

$$(\forall Z \in \Delta.\ [\![Z]\!]\eta = \mathit{true}) \text{ ならば } ([\![X]\!]\eta = \mathit{false} \text{ または } [\![Y]\!]\eta = \mathit{true})$$

であり，すなわち

$$(\forall Z \in \Delta.\ [\![Z]\!]\eta = \mathit{true}) \text{ ならば } [\![X \supset Y]\!]\eta = \mathit{true}$$

である．よって，$\Delta \models X \supset Y$ である．

$A = X \wedge Y$ の場合と $A = X \vee Y$ の場合は，$A = X \supset Y$ の場合と同様にして帰納法の仮定を用いて証明できる．

問 12.8(p. 162) 排中律が例の 1 つである：$A \vee (A \supset \bot)$

問 12.9(p. 166) 定理 12.2 は以下の通りである：

1. もし $\mathcal{N} \vdash \Delta \rhd A$ ならある M があって $\Lambda \vdash \Gamma_\Delta \rhd M : \tau_A$ である．
2. 逆に $\Lambda \vdash \Gamma_\Delta \rhd M : \tau_A$ なら $\mathcal{N} \vdash \Delta \rhd A_\tau$ である．

1 を $\mathcal{N} \vdash \Delta \rhd A$ の導出に関する帰納法によって証明する．$\mathcal{N} \vdash \Delta \rhd A$ の導出の最後に使用された（図 12.2 の）推論規則により，場合分けを行う．

(taut) の場合．$\Delta \cup A \rhd A$ であり，ある x と τ_A があって $\Gamma_{\Delta \cup A} = \Gamma_\Delta\{x : \tau_A\}$ であり，(var) より $\Gamma_{\Delta \cup A} \rhd x : \tau_A$ である．

(Ax) の場合．$\Delta \rhd a$ であり，命題定数 a に対応する定数を $c_a : \tau_a \in \mathit{Const}$ と仮定すると，(const) より $\Gamma_\Delta \rhd c_a : \tau_a$ である．

($\supset$:I) の場合．$\Delta \rhd A \supset B$ かつ $\Delta \cup A \rhd B$ である．帰納法の仮定より $\Delta \cup A \rhd B$ ならある M があって $\Gamma_{\Delta \cup A} \rhd M : \tau_B$ である．したがって，ある x と τ_A があって $\Gamma_{\Delta \cup A} = \Gamma_\Delta\{x : \tau_A\}$ であり，(abs) より $\Gamma_\Delta \rhd \lambda x.M : \tau_A \to \tau_B$ である．さらに，任意の A と B に対して，$\tau_{A \supset B} = \tau_A \to \tau_B$ である．

($\supset$:E) の場合．$\Delta \rhd B$, $\Delta \rhd A \supset B$, かつ $\Delta \rhd A$ である．帰納法の仮定より，$\Delta \rhd A \supset B$ ならある M_1 があって $\Gamma_\Delta \rhd M_1 : \tau_{A \supset B}$（すなわち，$\Gamma_\Delta \rhd M_1 : \tau_A \to \tau_B$）であり，$\Delta \rhd A$ ならある M_2 があって $\Gamma_\Delta \rhd M_2 : \tau_A$ である．このような M_1 と M_2 を仮定する．(app) より $\Gamma_\Delta \rhd M_1\ M_2 : \tau_B$ である．

【省略】以下同様にして，

- ($\wedge$:I) のとき (prod)
- ($\wedge$:E1) のとき (proj1)
- ($\wedge$:E2) のとき (proj2)
- ($\vee$:I1) のとき (inj1)

[*3] 無理数とは，有理数ではない実数のこと．

- (∨:I2) のとき (inj2)
- (∨:E) のとき (case)

を使用して示せばよい.

2 も同様に示せる．p. 164 l.-9– にあるように，「規則 (fix) は，各 τ に対して型 $(\tau \to \tau) \to \tau$ をもつ定数 $fix^{(\tau\to\tau)\to\tau}$ の存在を仮定」していることと，規則 (let) は規則 (abs) と規則 (app) を組み合わせることで同じ型をもつ式を構成できることに注意.

問 12.10(p. 168) 以下のラムダ式の定義の $\lambda x.M$ 以外の場合を示せば良い.

$$\begin{aligned} M ::= {} & c \mid x \mid \lambda x.M \mid M\ M \mid (M, M) \mid M[1] \mid M[2] \\ & \mid 1(M) \mid 2(M) \mid (\mathbf{case}\ M\ \mathbf{of}\ 1(x) \Rightarrow M, 2(x) \Rightarrow M) \\ & \mid \mathbf{let}\ x = M\ \mathbf{in}\ M \mid \mathbf{fix}(M) \end{aligned}$$

c の場合．仮定より $\Gamma \triangleright c : \tau$ かつ $y \notin dom(\Gamma)$ である．補題 12.5 より $\emptyset \triangleright c : \tau$ であり，さらに補題 12.4 より $[y/x]\Gamma \triangleright c : \tau$ である．したがって，$[y/x]\Gamma \triangleright [y/x]c : \tau$ である.

z の場合．仮定より $\Gamma \triangleright z : \tau$ かつ $y \notin dom(\Gamma)$ である．補題 12.3 より，$\Gamma \triangleright z : \tau$ であるので，$z \in dom(\Gamma)$ である．したがって，$z \neq y$ である．よって,

$$\begin{aligned} \Gamma \triangleright z : \tau &= \Gamma|\bar{x} \triangleright z : \tau && \because \text{補題 12.5} \\ &= \Gamma|\bar{x}\{y : \Gamma(x)\} \triangleright z : \tau && \because \text{補題 12.4} \\ &= [y/x]\Gamma \triangleright z : \tau && \because \text{p. 164 } [y/x]\Gamma \text{ の定義} \\ &= [y/x]\Gamma \triangleright [y/x]z : \tau && \because z = [y/x]z \end{aligned}$$

$M\ M$, (M, M), $M[1]$, $M[2]$, $1(M)$, $2(M)$, $\mathbf{fix}(M)$ の場合は帰納法の仮定を用いて簡単に示せる．($\mathbf{case}\ M\ \mathbf{of}\ 1(x) \Rightarrow M, 2(x) \Rightarrow M$) と $\mathbf{let}\ x = M\ \mathbf{in}\ M$ の場合は，束縛変数を $\lambda x.M$ の場合と同様に扱うことで示せる．【省略】

問 12.11(p. 169) x と $\lambda x.\ M$ 以外の残りの場合を示せば良い.

c の場合.

$$\begin{aligned} \Gamma\{x : \tau_1\} \triangleright c : \tau_2 &= \Gamma \triangleright c : \tau_2 && \because \text{補題 12.5} \\ &= \Gamma \triangleright [M_2/x]c : \tau_2 \end{aligned}$$

$M\ M$, (M, M), $M[1]$, $M[2]$, $1(M)$, $2(M)$, $\mathbf{fix}(M)$ の場合は帰納法の仮定を用いて簡単に示せる．($\mathbf{case}\ M\ \mathbf{of}\ 1(x) \Rightarrow M, 2(x) \Rightarrow M$) と $\mathbf{let}\ x = M\ \mathbf{in}\ M$ の場合は，束縛変数を $\lambda x.M$ の場合と同様に扱うことで示せる．【省略】

問 12.12(p. 169) ラムダ式の構造に関する帰納法を用いる.

命題 12.18 の証明.

c の場合.

$$\Gamma \triangleright \lambda x.\ c : \tau = \Gamma \triangleright \lambda y.\ [y/x]c : \tau$$

x の場合.

$$\begin{aligned} \Gamma \triangleright \lambda x.\ x : \tau &= \Gamma \triangleright \lambda y.\ y : \tau \\ &= \Gamma \triangleright \lambda y.\ [y/x]x : \tau \end{aligned}$$

z ($\neq x$) の場合.

$$\Gamma \triangleright \lambda x.\ z : \tau = \Gamma \triangleright \lambda y.\ [y/x]z : \tau$$

$\lambda z.M$ の場合．帰納法の仮定より $\Gamma \triangleright \lambda z.M : \tau_1 \to \tau_2$ かつ $y \notin FV(M)$ なら $\Gamma \triangleright \lambda y.[y/z]M : \tau_1 \to \tau_2$ である．補題 12.7 を使う．【省略】

命題 12.19 の証明.

c の場合．自明.

x の場合．自明.

$\lambda z.M$ の場合．仮定より $\Gamma \triangleright \lambda z.M : \tau_2 \to \tau_1$ かつ $\lambda z.M \equiv \lambda z.N$ である．$\Gamma \triangleright \lambda z.M : \tau_2 \to \tau_1$ が成り立つので，これが根にある導出木が存在する．図 12.3 より根の導出に用いられる規則は (abs) のみである．したがってその前提 $\Gamma\{z : \tau_2\} \triangleright M : \tau_1$ が成り立つ．帰納法の仮定（$\Gamma' \triangleright M : \tau_1$ かつ $M \equiv N$ なら $\Gamma' \triangleright N : \tau_1$）より，$\Gamma\{z : \tau_2\} \triangleright N : \tau_1$ である．(abs) より $\Gamma \triangleright \lambda z.N : \tau_2 \to \tau_1$ である.

$M_1\ M_2$ の場合．仮定より $\Gamma \triangleright M_1\ M_2 : \tau_1$ かつ $M_1\ M_2 \equiv N_1\ N_2$ である．式の構成法よりある τ_2 が存在して $\Gamma \triangleright M_1 : \tau_2 \to \tau_1$, $\Gamma \triangleright M_2 : \tau_2$, $M_1 \equiv N_1$, および $M_2 \equiv N_2$ である．帰納法の仮定（$\Gamma \triangleright M_1 : \tau_2 \to \tau_1$ かつ $M_1 \equiv N_1$ なら $\Gamma \triangleright N_1 : \tau_2 \to \tau_1$ であり，$\Gamma \triangleright M_2 : \tau_2$ かつ $M_2 \equiv N_2$ なら $\Gamma \triangleright N_2 : \tau_2$ である.）より，$\Gamma \triangleright N_1 : \tau_2 \to \tau_1$ かつ $\Gamma \triangleright N_2 : \tau_2$ である．(app) より，$\Gamma \triangleright N_1\ N_2 : \tau_1$ である.

【残りの場合は省略】

問 12.13(p. 170) p. 164 の図 12.3 を用いて導出木を作れば良い.

型環境を次のように定める：

$$\begin{aligned}\Gamma &= \{mul : int \times int \to int\}\\ \Gamma' &= \Gamma\{x : int\}\\ \Gamma'' &= \Gamma'\{y : int\}\\ \Gamma''' &= \Gamma''|_{\bar{x}}\{x : int \times int\}\end{aligned}$$

導出木 Π_1 を次のように定める：

$$\text{(app)}\ \frac{\Gamma''' \triangleright mul : int \times int \to int \qquad \text{(prod)}\ \dfrac{\text{(proj1)}\ \dfrac{\Gamma''' \triangleright x : int \times int}{\Gamma''' \triangleright x[1] : int} \qquad \text{(proj2)}\ \dfrac{\Gamma''' \triangleright x : int \times int}{\Gamma''' \triangleright x[2] : int}}{\Gamma''' \triangleright (x[1], x[2]) : int \times int}}{\text{(abs)}\ \dfrac{\Gamma''' \triangleright mul(x[1], x[2]) : int}{\Gamma'' \triangleright \lambda x.mul(x[1], x[2]) : int \times int \to int}}$$

求める導出木は以下の通り：

$$\text{(prod)}\ \frac{\text{(abs)}\ \dfrac{\text{(abs)}\ \dfrac{\text{(prod)}\ \dfrac{\text{(prod)}\ \dfrac{\Gamma'' \triangleright x : int \qquad \text{(app)}\ \dfrac{\Pi_1 \qquad \text{(prod)}\ \dfrac{\Gamma'' \triangleright x : int \qquad \Gamma'' \triangleright y : int}{\Gamma'' \triangleright (x, y) : int \times int}}{\Gamma'' \triangleright (\lambda x.mul(x[1], x[2]))(x, y) : int}}{\Gamma'' \triangleright (x, (\lambda x.mul(x[1], x[2]))(x, y)) : int \times int}}{\Gamma' \triangleright \lambda y.(x, (\lambda x.mul(x[1], x[2]))(x, y)) : int \to int \times int}}{\Gamma \triangleright \lambda x.\lambda y.(x, (\lambda x.mul(x[1], x[2]))(x, y)) : int \to int \to int \times int} \qquad \Gamma \triangleright 2 : int}{\Gamma \triangleright (\lambda x.\lambda y.(x, (\lambda x.mul(x[1], x[2]))(x, y)))\ 2 : int \to int \times int} \qquad \Gamma \triangleright 3 : int}{\Gamma \triangleright (\lambda x.\lambda y.(x, (\lambda x.mul(x[1], x[2]))(x, y)))\ 2\ 3 : int \times int}$$

2 つ目の λx の内側に現れる $x[1]$ と $x[2]$ は型環境 Γ''' の中の x の型 $int \times int$ で型チェックされ，内側の λx の定義に対応した変数として扱われている.

問 12.14(p. 173) 定理 12.11 において N_1 と N_2 が正規形である場合を考えればよい.

$M \Downarrow N_1$ かつ $M \Downarrow N_2$ と仮定する．したがって，$M \xrightarrow{*} N_1$ かつ $M \xrightarrow{*} N_2$ である．定理 12.11 より，$N_1 \xrightarrow{*} N_3$, $N_2 \xrightarrow{*} N_4$ かつ $N_3 \equiv N_4$ となる N_3, N_4 が存在する．N_1 と N_2 が正規形であることから，$N_1 = N_3$ かつ $N_2 = N_4$ である[*4]．したがって，$N_1 \equiv N_2$ である.

問 12.15(p. 178) η 簡約が含まれる反例を考えると良い.

[*4] ここでは N_1 と N_3, N_2 と N_4 がそれぞれ完全に一致しているので “=” を用いた．なお，$\equiv$ は α 同値性を表す記号である.

任意のモデル $\mathcal{A}$ について,

$$
\begin{aligned}
&\mathcal{A}[\![\Gamma \triangleright \lambda x.\ inc\ x : int \to int]\!]\eta \\
&= v \in A^{int} \text{に } \mathcal{A}[\![\Gamma\{x : int\} \triangleright inc\ x : int]\!]\eta\{x : v\} \text{ を対応させる関数} \\
&= v \in A^{int} \text{に } \mathcal{A}[\![\Gamma\{x : int\} \triangleright inc : int \to int]\!]\eta\{x : v\}(\mathcal{A}[\![\Gamma\{x : int\} \triangleright x : int]\!]\eta\{x : v\}) \text{ を対応させる関数} \\
&= v \in A^{int} \text{に } \overline{inc}(\mathcal{A}[\![\Gamma\{x : int\} \triangleright x : int]\!]\eta\{x : v\}) \text{ を対応させる関数} \\
&= v \in A^{int} \text{に } \overline{inc}(v) \text{ を対応させる関数} \\
&= \overline{inc} \\
&\mathcal{A}[\![\Gamma \triangleright inc : int \to int]\!]\eta
\end{aligned}
$$

である．したがって,

$$\models \{inc : int \to int\} \triangleright \lambda x.\ inc\ x = inc : int \to int$$

である．しかし，2 つの異なるラムダ式 inc, $\lambda x.\ inc\ x$ は，図 12.4 のどの変形規則でも変形することができない．したがって

$$\vdash \{inc : int \to int\} \triangleright \lambda x.\ inc\ x = inc : int \to int$$

ではない.

問 12.16(p. 179) ヒントにあるとおりに，M の構造に関する帰納法を用いる．

x の場合．型システムの定義より $\tau_1 = \tau_2$ である．両辺が $\mathcal{A}[\![\Gamma \vdash N : \tau_1]\!]\eta$ になり成立する．

y $(y \neq x)$ の場合．両辺が $\eta(y)$ となり成立する．

$\lambda y.M_0$ の場合．§12.2 で説明した束縛変数に関する約束を仮定する．自明.【省略】

問 12.17(p. 179)

(fst) の場合．$\Lambda \vdash \Gamma \triangleright (M_1, M_2)[1] : \tau_1$ と仮定する．型システムの定義よりある τ_2 が存在して，$\Lambda \vdash \Gamma \triangleright (M_1, M_2) : \tau_1 \times \tau_2$ であり，$\Lambda \vdash \Gamma \triangleright M_1 : \tau_1$ かつ $\Lambda \vdash \Gamma \triangleright M_2 : \tau_2$ である．η を Γ を満たす任意の $\mathcal{A}$ 環境とする．以下の等式が成り立つ.

$$
\begin{aligned}
&\mathcal{A}[\![\Gamma \triangleright (M_1, M_2)[1] : \tau_1]\!]\eta \\
&= Proj_1^{\tau_1,\tau_2}(\mathcal{A}[\![\Gamma \triangleright (M_1, M_2) : \tau_1 \times \tau_2]\!]\eta) && \because \text{p. 177 図 12.6 l.8} \\
&= Proj_1^{\tau_1,\tau_2}(Prod^{\tau_1,\tau_2}(\mathcal{A}[\![\Gamma \triangleright M_1 : \tau_1]\!]\eta, \mathcal{A}[\![\Gamma \triangleright M_2 : \tau_2]\!]\eta)) && \because \text{p. 177 図 12.6 l.6–7} \\
&= \mathcal{A}[\![\Gamma \triangleright M_1 : \tau_1]\!]\eta && \because \text{p. 175 (ii)}
\end{aligned}
$$

以上より，$\models \Gamma \triangleright (M_1, M_2)[1] = M_1 : \tau_1$ が成り立つ.

【省略】

問 12.18(p. 181) (i) 補題 12.16(i) を示す．$\Gamma \triangleright (\lambda x.M)\ N : \tau$ と仮定する．これを根とする導出木が存在するので，p. 164 図 12.3 の (app) と (abs) よりある τ_1 が存在して $\Gamma\{x : \tau_1\} \triangleright M : \tau$ と $\Gamma \triangleright N : \tau_1$ である．補題 12.7(p. 168) より $\Gamma \triangleright [N/x]M : \tau$ である.

【省略】

(ii) 簡約関係の回数に関する帰納法で示すと良い.

問 12.19(p. 182) M_2 の型付けに失敗する場合がある.

$\Lambda \vdash \emptyset \triangleright 3 : int$ かつ $(\lambda x.3)\ (1\ 2) \xrightarrow{*} 3$ であっても $\Lambda \vdash \emptyset \triangleright (\lambda x.3)\ (1\ 2) : int$ とならない.

$\Lambda \vdash \emptyset \triangleright 3 : int$ かつ $(3, x)[1] \xrightarrow{*} 1$ であっても $\Lambda \vdash \emptyset \triangleright (3, x)[1] : int$ とならない．図 12.3(p. 164) の型導出システムでは型を導出することができないからである.

（構文論的性質）すべての変数と定数に型を注釈として付記する．関数適用などでは型が整合していないと正しいラムダ式では無いものとする．(cf. 単純型付けラムダ式)

問 12.20(p. 183) (fst) に対応する証明の簡略変換は

$$\dfrac{\dfrac{\begin{matrix}\Pi_1 \\ (\Delta \triangleright A)\end{matrix} \qquad \begin{matrix}\Pi_2 \\ (\Delta \triangleright B)\end{matrix}}{\Delta \triangleright A \wedge B}\,(\wedge\text{:I})}{\Delta \triangleright A}\,(\wedge\text{:E1})$$

から

$$\begin{matrix}\Pi_1 \\ (\Delta \triangleright A)\end{matrix}$$

への変換である．(snd) に対応する証明の簡略変換は

$$\dfrac{\dfrac{\begin{matrix}\Pi_1 \\ (\Delta \triangleright A)\end{matrix} \qquad \begin{matrix}\Pi_2 \\ (\Delta \triangleright B)\end{matrix}}{\Delta \triangleright A \wedge B}\,(\wedge\text{:I})}{\Delta \triangleright B}\,(\wedge\text{:E2})$$

から

$$\begin{matrix}\Pi_2 \\ (\Delta \triangleright B)\end{matrix}$$

への変換である．(case1) に対応する証明の簡略変換は

$$\dfrac{\dfrac{\begin{matrix}\Pi_1 \\ (\Delta \triangleright A)\end{matrix}}{\Delta \triangleright A \vee B}\,(\vee\text{:I1}) \qquad \begin{matrix}\Pi_2 \\ (\Delta \cup A \triangleright C)\end{matrix} \qquad \begin{matrix}\Pi_3 \\ (\Delta \cup B \triangleright C)\end{matrix}}{\Delta \triangleright C}\,(\vee\text{:E})$$

から

$$\dfrac{\begin{matrix}\Pi_1 \\ (\Delta \triangleright A)\end{matrix} \qquad \dfrac{\begin{matrix}\Pi_2 \\ (\Delta \cup A \triangleright C)\end{matrix}}{\Delta \triangleright A \supset C}\,(\supset\text{:I})}{\Delta \triangleright C}\,(\vee\text{:E})$$

への変換である．(case2) に対応する証明の簡略変換は

$$\dfrac{\dfrac{\begin{matrix}\Pi_1 \\ (\Delta \triangleright B)\end{matrix}}{\Delta \triangleright A \vee B}\,(\vee\text{:I1}) \qquad \begin{matrix}\Pi_2 \\ (\Delta \cup A \triangleright C)\end{matrix} \qquad \begin{matrix}\Pi_3 \\ (\Delta \cup B \triangleright C)\end{matrix}}{\Delta \triangleright C}\,(\vee\text{:E})$$

から

$$\dfrac{\begin{matrix}\Pi_1 \\ (\Delta \triangleright B)\end{matrix} \qquad \dfrac{\begin{matrix}\Pi_3 \\ (\Delta \cup B \triangleright C)\end{matrix}}{\Delta \triangleright B \supset C}\,(\supset\text{:I})}{\Delta \triangleright C}\,(\vee\text{:E})$$

への変換である．

参考文献

[1] 高橋陽一郎ほか：数学 I，啓林館 (2011).

[2] 浅井健一：プログラミングの基礎，サイエンス社 (2007). サポートページ：`http://pllab.is.ocha.ac.jp/~asai/book/Top.html`.

[3] RobertMecklenburg（著），矢吹道郎（監訳），菊池彰（訳）：GNU Make 第 3 版，O'Reilly (2005). Available from `https://www.oreilly.co.jp/library/4873112699/`.

[4] Milner, R., Tofte, M. and Harper, R.: *The Definition of Standard ML (revised*, MIT Press (1997). Available from `https://smlfamily.github.io/`.

[5] Paulson, L.C.: *ML for the Working Programmer (2nd Ed.)*, Cambridge University Press (1996). Available from `https://www.cl.cam.ac.uk/~lp15/MLbook/`.

[6] Ullman, J.D.: *Elements of ML Programming*, Prentice-Hall, Inc. (1994). Supplemental materials available from `http://infolab.stanford.edu/~ullman/emlp.html`.

[7] Cousineau, G. and Mauny, M.: *The Functional Approach to Programming*, Cambridge University Press (1998).