

1 集合論の基礎

問題 1.1 上記の性質が、前述の順序対の集合による表現の上で、成り立つことを証明せよ。

証明. 順序対 (a, b) を集合 $\{\{a\}, \{a, b\}\}$ と取ることができるから

$$(a, b) = (a', b') \iff \{\{a\}, \{a, b\}\} = \{\{a'\}, \{a', b'\}\}$$

したがって、 $(a, b) = (a', b')$ のとき

$$\{a\} \in \{\{a'\}, \{a', b'\}\} \ \& \ \{a, b\} \in \{\{a'\}, \{a', b'\}\}$$

を得る。 $\{a\} \in \{\{a'\}, \{a', b'\}\}$ より

$$\{a\} = \{a'\} \text{ or } \{a\} = \{a', b'\}$$

同様に、 $\{a, b\} \in \{\{a'\}, \{a', b'\}\}$ より

$$\{a, b\} = \{a'\} \text{ or } \{a, b\} = \{a', b'\}$$

(i) $\{a\} = \{a', b'\}$ のとき、 $a = a' = b'$ である。このとき、 $\{a, b\} = \{a'\}$ と $\{a, b\} = \{a', b'\}$ は同等である。 $\{a, b\} = \{a', b'\}$ のときを考えて、 $a = b = a' = b'$ を得る。したがって、 $a = a' \ \& \ b = b'$ が成り立つ。

(ii) $\{a\} = \{a'\}$ のとき、 $a = a'$ である。 $\{a, b\} = \{a'\}$ のとき、 $a = b = a'$ で、 $\{a\} = \{a', b'\}$ より $\{b\} = \{b, b'\}$ となるから、 $b = b'$ を得る。したがって、 $a = a' \ \& \ b = b'$ が成り立つ。また、 $\{a, b\} = \{a', b'\}$ のとき、 $a = b' \text{ or } b = b'$ である。 $a = b'$ のとき、 $a = a' = b'$ となるから (i) と同等の議論で $a = a' \ \& \ b = b'$ が成り立つ。 $b = b'$ のとき、まさに $a = a' \ \& \ b = b'$ が成り立つ。

(i), (ii) より

$$(a, b) = (a', b') \implies a = a' \ \& \ b = b'$$

逆に、 $a = a' \ \& \ b = b' \implies (a, b) = (a', b')$ が成り立つ。

$$\therefore (a, b) = (a', b') \iff a = a' \ \& \ b = b'$$

■

問題 1.2 X と Y が集合であるとき、 $X \multimap Y$ と $X \rightarrow Y$ はどちらも集合である。

解答. そうですね。

問題 1.3 $R \subseteq X \times Y$ かつ $S \subseteq Y \times Z$ かつ $T \subseteq Z \times W$ であるとする。 $T \circ (S \circ R) = (T \circ S) \circ R$ であること (すなわち、合成が結合的 (associative) であること) を確かめよ。また、 $R \circ Id_X = Id_Y \circ R = R$ であること (すなわち、恒等関数は合成操作に関する恒等元であること) を確かめよ。

証明.

■

2 入門：操作的意味論

問題 2.1 もしプログラミング言語 ML (OCaml, SML) や Haskell でプログラミングができるのであれば, IMP の構文をデータ型として定義せよ. さらに, 構文要素 e_0, e_1 に対してその同一性 $e_0 \equiv e_1$ を判定するプログラムを書け.

解答. `imp.hs` を参照. 実行例を以下に示す. (➡注 純粋関数型言語 Haskell で記述しています.)

```
ghci> aexpEqual (Add (Const 3) (Const 5)) (Add (Const 5) (Const 3))
Fal

ghci> aexpEqual (Add (Const 3) (Const 5)) (Add (Const 3) (Const 5))
Tru

ghci> bexpEqual Tru (Not (Not Tru))
Fal
```

問題 2.2 算術式の評価規則をプログラミングせよ. 問題 2.1 で導入したデータ型を使うとよい.

解答. `imp.hs` を参照. 実行例を以下に示す.

```
ghci> evalAexp ((Mul (Add (Const 2) (Const 6)) (Sub (Const 7) (Const 2))), states)
40
```