論理とプログラム
仕様記述言語 Z にしたがって

八杉満利子*

平成19 年 4 月 16 日

目 次

*京都産業大学理学部コンピュータ科学科専門科目・理学部共通教養専門科目

1 序文

以下の内容は主として

参考書１： Z An Introduction to Formal Methods, Second Edition, by Antoni Diller, John Wiley and Sons, 1994;

参考書２： ソフトウエア仕様記述の先進技法-Z 言語」（B. ポター他著、田中監訳、プレンティスホール/トッパン）

の一部分を参考にしています。

注意：ごく一部分の参照であること、直訳ではないこと、多くの部分が担当者オリジナルであること、この講義のテキストとしてのみ使用すること、などの事情により参考書の著者・出版社の許可はとっていません。したがってこの内容の一部または全体をこの講義のテキスト以外の目的で使用しないでください。自分用に余分のコピーを作ることはかまいませんが、大量にコピーをとったり、この科目の受講者以外に配布することはしないでください。一般に著作権は非常にデリケートな問題で、法的責任問題になることもありますので、注意してください。

この講義の主目的は命題論理・述語論理に習熟し、将来ソフトウエア科学・技術に使えるようにすることです。しかし論理は数学、自然科学、情報科学、のみではなく、広く人文・社会科学、法律、工学などにおいても基本になるものであり、また日常生活でも必要なものです。その意味ではだれでも基本的な論理体系である命題論理・述語論理の基本は知っておくべきなのです。この講義は受講者各人の目的意識をもって聞けるようにします。

命題論理はできるだけ簡潔にします。詳しいことは八杉のホームページの「集合と論理２００７」のサイトを参考にしてください。「集合と論理」のテキスト「情報系の数学入門」（林・八杉著、オーム社）も参考になります。

ここでは主点は述語論理におきます。命題論理は述語論理の一部分なのです。述語論理の初歩は「情報系の数学入門」の最後にあります。述語論理を一般的に学ぶよりも、それが実用化されている場面で学ぶほうが興味がもてると思います。そのために、最初から仕様記述言語 Z に沿って述語論理の説明をしていきます。

言語 Z とは何か、どのような効用があるのか、については上記参考書に詳しく書いてありますが、八杉特研の卒業研究に簡潔に述べられています。この科目のページにおいてありますから、参考にしてください。（これももちろん著作権は作者たちにあります。）

「仕様」とは、プログラム作成にあたって、そのプログラムにしてほしい作業内容です。非常に簡単な例では、”「二つの数の最大公約数を求める」プログラムを書いてください ”と依頼されたときの「 $\cdots$ 」の内容のことです。しかし、依頼者は暗黙のうちにすべての整数の組について最大公約数を求め

たいのに、プログラマは正整数だけと思い込んでプログラムを書くかもしれません。できあがったプログラムを依頼者が動かそうとして 0 や負の数を入力したらどうなるか？このような状況を避けるために、プログラムの条件をもっと詳しく書いておくほうがいいでしょう。そのようにプログラムに要求する条件を正確に書いておけば行き違いは減ります。銀行のオンラインシステム、交通システム、原子力発電所の管理、などのプログラムならば、ちょっとした思い違いが混乱や惨事を引き起こしかねません。そのためにも作成してほしいプログラムの条件を詳細に正確に書いておく必要があります。そうしておけば、依頼者と直接接触しない開発者でも自信をもってプログラム開発できるでしょう。そのような条件がプログラムの仕様記述であり、それをできるだけ正確に書くためには形式を整えなくてはなりません。使ってよい記号や書き方を約束事として指定できるのです。つまり仕様の規格化をするのです。その一つで成功し広く使われているのが、イギリスで開発された Z 言語です。Z の基本は述語論理による集合の記法と、述語論理の駆使です。Z のファイルの作り方についてはインターネットで Z のサイトを探してください。とくに

http://vl.zuser.org/

に詳しく載っています。ぜひ一度見てください。

2　命題論理：Propositional calculus (logic)

2.1　論理式：命題論理の場合

「論理式」は formula の訳です。命題とは概念としての文章です。文字で書いたり発声したりすると文章になる、そのもとの考え、というべきものです。命題をことばで表現するのが文章です。文章の接続詞によるつながり具合、つまり文章の論理的構造にのみ着目し、文章が成り立つ（真である）か、成り立たない（偽である）か、のみを問うのが命題論理です。日本語とか英語というような言語に関わらず一定の規約によってだれでも読めるようにしたものが命題論理の論理式です。

この節では論理式といえば命題論理の論理式なので、簡単に「論理式」と呼びます。文章はそれ自体が真であるか偽であるようなそれ以上分解できない基本的な文章から、「そして」とか「ならば」などの接続詞でつないで作られます。たとえば「海王星は地球よりも太陽から遠い」や $3 \geq 9$ は基本命題です。前者は正しいですが、後者は明らかに偽です。「海王星は地球よりも太陽から遠く、（そして）$3 \geq 9$」は意味のない文章ではあっても文法的には認められるものであり、真な文章と偽な文章を「そして」でつないでいるので全体としては偽です。このような文章（命題）は複合文（命題）と呼びます。

命題（文章）を記号で表すものが論理式です。多くの場合、ローマ字の大文

字で表します。とくに基本命題を表す文字を基本論理式と呼び、それらをつないで複合文を表す論理式を構成する、「つなぎ」を論理結合子、命題結合子、（命題）論理記号などと呼びます (logical connective, propositional connective, lgical symbol).

それらは $\neg, \wedge, \vee, \Rightarrow$ の４種類です。それぞれ「 … でない」、「そして」、「または」、「ならば」と読み、またそういう意味を表します。文章のつなぎなんていくらでもあるのに、と思われるかもしれません。でも論理的にはこの４個で十分なのです。実際、十分性が証明できるのです。さて、論理式の定義。

定義 2.1 （論理式：formula） BNF 記法による。

$$A ::= ident|\neg A|A_1 \wedge A_2|A_1 \vee A_2|A_1 \Rightarrow A_2$$

A は一般に論理式を表す。$ident$ は任意のローマ字または数字つきのローマ字による識別子であり、基本論理式を表す。上の記法は次のように読む。

“論理式は基本論理式か、すでに定義された論理式から４個の論理結合子で合成されたもの、である。“

基本論理式を命題変数とも呼ぶ。

命題変数という名は、たとえば実数を表す変数 $x, y, \cdots$ などと同じ意味をもつ。$x, y, \cdots$ は「実数を表すもの」というだけでそれ自体特別な値をもたない。$x = 3.1$ というように値を付与できる、という働きをもつ。同様に、命題変数 P は「命題を表すもの」であり、たとえば「P は ”今日は月曜日である ”という命題を表すものとする」と指定すれば、P に特別な内容を付与できる。複合（合成）論理式を作ることは $x + y$ というように変数から式を作ることに相当する。

命題論理の論理式の例は「集合と論理２００７」および「情報系の数学入門」にたくさんあるので、参照してください。

2.2 真理値:：truth value

定義 2.2 （真理値） 1) 各論理式 A に真理値と呼ばれる値を対応させる。その対応（v と書き、付値と呼ぶ）は論理式の集合から真理値の集合 $\{t, f\}$ への写像である。t は真、f は偽を表す。

基本論理式にはその都度自分で値を与える。複合論理式は基本論理式の値と論理結合子の意味から決まる。正確な定義は次のとおりである。

$\leftrightarrow$ は、「左辺が成り立つのはちょうど右辺が成り立つときである」を略記している。すなわち、$A \leftrightarrow B$ は、A が成り立つことと B が成り立つことが同値であることを表す。

任意の論理式 A について、$v(A)$ は t か f か、ちょうど 1 個の値をとる。X は基本論理式を表すとする。

$$v(X) \in \{t, f\}$$

（$v(X)$ は集合 $\{t, f\}$ の要素である。すなわち、t か f である。どちらであるか、は任意である。）

$$v(\neg A) = t \leftrightarrow v(A) = f$$
$$v(A \land B) = t \leftrightarrow v(A) = t \quad \text{かつ} \quad v(B) = t$$
$$v(A \lor B) = t \leftrightarrow v(A) = t \quad \text{または} \quad v(B) = t$$
$$v(A \Rightarrow B) = t \leftrightarrow v(A) = t \quad \text{のときには} \quad v(B) = t$$

最後の右辺について、（日本語の論理的な意味から）$v(A) = t$ でないときには $v(B)$ の値には何も要請していない、すなわち任意であること、に注意。

この定義を見やすくするために真理値表というものを使うのがふつうである。

定義 2.3 （真理値表） $\neg$ の真理値表

P	$\neg P$
t	f
f	t

$\land$ の真理値表

P	Q	$P \land Q$
t	t	t
t	f	f
f	t	f
f	f	f

$\lor$ の真理値表

P	Q	$P \lor Q$
t	t	t
t	f	t
f	t	t
f	f	f

P	Q	$P \Rightarrow Q$
t	t	t
t	f	f
f	t	t
f	f	t

$P \Rightarrow Q$ は $\neg P \vee Q$ と同値（真理値の意味で）である。自分で確かめてみるとよい。また、内容的には $\neg(P \wedge \neg Q)$ と考えると、分かり易いかもしれない。

少しだけ例題：

(1) $v(A \vee \neg A) = t$（恒に真、恒真）

$v(A) = t \leftrightarrow v(\neg A) = f$ であるから、$v(A)$ が t でも f でも $v(A), v(\neg A)$ のどちらかが t になる。したがって $\vee$ の真理値の定義から、$v(A \vee \neg A)$ は恒に t。

(2) $v(X \Rightarrow \neg X) = v(\neg X)$。ゆえに $v(X \Rightarrow \neg X) = t \leftrightarrow v(X) = f$。（恒真でないが、充足可能）

(3) $v(\neg(P \Rightarrow P)) = f$（恒に f、すなわち矛盾）。

3　述語論理：predicate calculus (logic)

述語論理は命題論理に、個体について言及する記号を加えたものです。命題論理は基本文を接続詞でつないだときの文の構成のみを問題にしていますが、それでは個体間の関係は表現できません。ふつう述語論理は個体、個体間の基本的な関係、不特定の個体の存在と全称についての言及によって構成されますが、ここでは言語 Z に沿った記法を採用するので、まず「型」の導入から始めます。「型」（type）はプログラム言語にあるデータ型と実質同じものです。整数型、実数型、リスト型、などがありますね。型はデータの性質ですが、それはまた全体集合ともみなせます。たとえば、整数型とは整数の集合

$$\mathbf{Z} = \{0, \pm 1, \pm 2, \cdots, \pm 100, \cdots\}$$

と考えられます。Mathematica で $f[x\_Integer] := 2x + 1$ というコードは x に整数を入力したとき $f[x]$ の値が $2x + 1$ である、ということを表します。$x\_Integer$ は $x \in \mathbf{Z}$ と同じ意味です。記法のちがいだけです。

3.1　変数：variable

変数は $x, y, z, \cdots, ca, person, \cdots$ など、小文字のアルファベットで表すのが慣例である。変数は述語論理では不特定の個体を表す。通常数学で $x, y, z, \cdots$ などが不特定の実数を表すのと同じ使い方である。数学では暗黙裡に x は実数とする、とか n は整数とする、などと仮定している。ここではそれを明示的に表現するために次の「型」を使う。

3.2　型:type

型とは、個体の集まり、つまり集合です。変数が値をとる範囲を指定します。集合演算における全体集合のようなものです。

Z の特性の一つはそれが型付言語 (typed language) ということである。ある変数が仕様ドキュメントで最初に導入されるとき、その型が与えられなければならない。

$\mathbf{Z}$（整数型）、$\mathbf{R}$（実数型）は整数全体の集合で、それがどのようなものか誰でも知っているが、一般には型の定義から始めなければならない。なお、Mathematica ではこれらがそれぞれ $Integer$, $Real$ で表現されている。
例：$[Europe]$ と書いて、ヨーロッパの国の集合であることを記す。こうして $Europe$ を基本型として使う。

3.3 宣言:declaration

個体 x が整数である（型 $\mathbf{Z}$ の個体である）ことを $x : \mathbf{Z}$ と表現する。”x は整数である ”または ”x の型は $\mathbf{Z}$ である ”などと読む。

例： $france : Europe \quad i, j, k : \mathbf{Z}$

$france, italy, poland : Europe; i, j, k : \mathbf{Z}$

; は、複数の型関係を列記するときに使う。

以上のような形式を「変数の型宣言」と呼ぶ。ある変数が指し示す（値としてとる）対象を指定する、という意味である。たとえば、$i : \mathbf{Z}$ は、「変数 i は整数を値としてとるものである」ことを表す。

3.4 述語記号：predicate (predicate symbol)

述語記号は個体間の関係を表す記号である。たとえば $=$ は 2 個体が等しいことを表す述語記号である。$=$ は $a = b$ のように記号の左右に個体または個体を表す変数を書くが、一般には次のように使用する。たとえば P が 2 個体間の関係であるとき、$P(x, y)$ は、x と y が P という関係にある、ということを表す。この書き方を $=$ に適用すれば $=(a, b)$ となる。この場合 P あるいは $=$ は「２引数述語である」という。P としては、大小関係 $<, >$ や友人関係、親子関係、ある国とその首都、などいろいろ例がある。

同じように R が 3 引数述語であれば、3 個体が R という関係にあることは $R(x, y, z)$ と表される。たとえば $a < b < c$ は 3 個の実数 a, b, c の大小関係を表すが、この関係を R で表すものとすると、$R(a, b, c)$ となる。

$P(a, b)$ や $R(a, b, c)$ などを述語論理における原子論理式（atomic formula）と呼ぶ。とくに命題変数は引数 0 の述語記号であり、それ自身原子論理式である。また、一つの個体についての性質も述語記号で表す。たとえば「n は偶数である」を $Even(n)$ と表すことにすると、$Even$ は 1 引数の述語記号である。

3.5 量記号: quantifier

個体に言及するケースとしては、「すべての個体について」（全称）と、「ある個体について」（存在）がある。「すべての実数について」、「世界中の国々について」などが全称的言及であり、「ある実数について」、「ある国について」は存在的言及である。

全称の例：(1) すべての実数 x について、$x^2 \geq 0$。

(2) どの国にも国旗がある。

存在の例：(3) $x < 0$ となる実数がある（負の実数がある）。

(4) 人口が 1 億を超える国がある。

(1) は変数 x がとる値の範囲が実数である、と指定している。これを (a)「すべての x について $x^2 \geq 0$」と書いて、この文が実際に成り立つかどうかを考えるときに、別途「ただし x は実数である」と指定することもできる。また、(1) の文について (b)「すべての x について、x が実数ならば、$x^2 \geq 0$ である」と書き換えることもできるし、(c)「すべての実数の型をもつ x について、$x^2 \geq 0$ である」と書くこともできる。一般に述語論理では (a) か (b) を採用するが、ここでは言語 Z にしたがって (c) を採用する。(c) を記号で書くと、

$$\forall x : \mathbf{R} \bullet x^2 \geq 0 \tag{1}$$

となる。ただし $\mathbf{R}$ は実数の型（実数の集合：Mathematica の Real に相当）、$\forall$ は「すべて」、$\bullet$ は括弧の役割をもつ。括弧を使えば

$$\forall x : \mathbf{R}(x^2 \geq 0)$$

と表される。どちらでもよい。

3.6 論理式：formula

(1) のような式を（Z 形式の）述語論理の論理式（formula）と呼ぶ。以下では「論理式」は述語論理の論理式を指す。

述語論理の論理式は、命題論理の論理式の作り方および限量記号の使用により構成される。たとえば A が論理式で X が型のとき、$\forall x : X \bullet A$, $\exists x : X \bullet A$ は論理式である。あるいは $\forall x : X(A)$, $\exists x : X(A)$ でもよい。

述語論理の論理式は「情報系数学入門」の pp.130-133 に詳細に定義してあるので、参照してほしい。

$\forall x$ を（x を変数とする）「全称記号」と呼ぶ。とくに (1) では型によって変数の範囲を指定するので、「限定された全称記号」と呼ぶ。宣言 $x : \mathbf{R}$ に現れる（出現する、という）x を「束縛する出現」（binding occurrence）と呼び、$\bullet$ 以後の x の出現を「束縛された出現」（bound occurrence）と呼ぶ。$\forall x : \mathbf{R}$ で束縛されているからである。

同様に、「負の実数がある」を論理式で書くと、

$$\exists x : \mathbf{R} \bullet x < 0 \tag{2}$$

または

$$\exists x : \mathbf{R}(x < 0)$$

となる。$\exists x$ は「x が存在する」、「ある x について」などと読み、$\exists$ は「存在記号」という。出現、限定、束縛、などは上と同じである。

全称記号（universal quantifier）と存在記号（existential quantifier）をあわせて「量記号」、「限量記号」（quantifier）などと呼ぶ。

なお、変数の型に加えてさらに範囲を限定する（全体集合の中の部分集合に相当）ときには、型宣言の後に | を書く。以下の例で説明する。

例 (i) $\forall x : \mathbf{Z} | x \in \{1,2,3,4\} \bullet x < 5$

任意の整数で集合 $\{1,2,3,4\}$ の要素であるものは、5 より小さい。

(ii) $\forall x : Europe | borders(x, albania) \bullet borders(x, bulgaria)$

ヨーロッパの国でアルバニアと国境を接する国はブルガリアと国境を接する。

これは次の論理式と同値である。

$$\forall x : Europe(borders(x, albania) \Rightarrow borders(x, bulgaria))$$

(iii) $\exists x : \mathbf{Z} | 0 \leq x \leq 5 \bullet x = x * x$

5 以下の負でない整数で、その 2 乗と等しいものがある。

これは次の論理式と同値である。

$$\exists x : Europe \bullet 0 \leq x \leq 5 \wedge x = x * x$$

(iv) $\exists x : Europe | borders(x, albania) \bullet isinEU(x)$

ヨーロッパの国でアルバニアと国境を接して EU の構成員である国がある。

(v) $Human$ は人類（人々の集合としての型）を表し、$Mortal(x)$ は「x は死ぬものである」（不死身ではない）という x の性質を表すものとする。

$$\forall x : Human \bullet \ Mortal(x)$$

およそ人は死ぬものである。

(vi) $Flower, L(x,y)$ はそれぞれ、花の集合（型）、「x は y を好きだ」を表すとする。

$$\forall x : Human \bullet \exists y : Flower \bullet L(x,y)$$

誰でも何か花が好きだ；人には誰でも好きな花がある。

注 1) $x < 5$, $borders(x, albania)$, $isinEU(x)$, $Mortal(x)$, $L(x,y)$ はそれぞれ 1 引数または 2 引数の述語記号 $<, borders, isinEU, Mortal, L$ と変数 x, y から作られる原子論理式である。

3.7 一般の述語論理

一般の述語論理では型を使わないので、上の例は型の代わりに述語で制限を表す。「情報系の数学入門」3・6（pp127-130）を参照。

例 (i) $\forall x(Integer(x) \wedge x \in \{1,2,3,4\} \Rightarrow x < 5)$

（$Integer(x)$ は「x は整数である」を表す述語とする。）

(ii) $\forall x(Europe(x) \wedge ba(x) \Rightarrow bb(x))$

(iii) $\exists x(Integer(x) \wedge 0 \leq x \leq 5 \wedge x = x * x)$

(iv) $\exists x(Europe(x) \wedge ba(x) \wedge eu(x))$
(v) $\forall x(Human(x) \Rightarrow Mortal(x))$
(vi) $\forall x(Human(x) \Rightarrow \exists y(Flower(y) \wedge L(x, y)))$

3.8 ユニークな存在

ある条件が成り立つような個体がある、というときには、その条件を満たす個体がたくさんあるかもしれない。しかしたとえば「二乗しても値が変わらない正整数がある」というときにその数は 1 以外にはない。すなわち条件を満たす個体はたった一つ、ユニークに決まる。ユニークな存在を $\exists_1$ で表す。

例 $\exists_1 x : \mathbf{Z} | x > 0 \bullet x = x * x$

$\exists_1$ を使わなければ、これは次のように表される。

$$\exists x : \mathbf{Z} | x > 0 \bullet x = x * x \wedge \forall y : \mathbf{Z}(y > 0 \wedge y = y * y \Rightarrow x = y)$$

または

$$\exists x : \mathbf{Z} \bullet x > 0 \wedge x = x * x \wedge \forall y : \mathbf{Z} \bullet y > 0 \wedge y = y * y \Rightarrow x = y$$

例

$$\exists_1 y : Human | inPotters(y) \bullet girl(x)$$

（ポッター家には女の子がただ一人いる）

$[Potters]$ によって Potters を型とすれば $\exists_1 y : Potters \bullet girl(y)$

注 「ユニーク」（unique）とは、「ちょうど一つ」という意味である。よく日本語で使われる、個性的とか変わっているとかいうような意味ではない。独特の、という意味もあるが、他にない、という意味で、やはりただ一つ、という意味合いが強い。

3.9 同値な論理式

Z においても通常の命題論理と同様に真理値について同値な論理式がある。$v(A \Rightarrow B) = v(\neg A \vee B)$ はその一例である。このように同値な論理式はその都度都合のよいほうを採用すればよい。量記号については次の同値性が成り立つ。

$$v(\forall x A(x)) = v(\neg \exists x \neg A(x))$$

$$v(\exists x A(x)) = v(\neg \forall x \neg A(x))$$

型つきの論理式では次のような同値性が成り立つ。同値なことを真理値関数を使わずに $\leftrightarrow$ で表す。

$$(\forall D | P \bullet Q) \leftrightarrow (\forall D \bullet P \Rightarrow Q)$$

$$(\exists D | P \bullet Q) \leftrightarrow (\exists D \bullet P \wedge Q)$$

ただし $\forall D$ は $\forall x : \mathbf{Z}$ のように量記号と型宣言の合成の略記。$\exists D$ も同様である。

例 (ii) $\forall x : Europe | b(x,a) \bullet b(x,b)$

$$\leftrightarrow \forall x : Europe \bullet b(x,a) \Rightarrow b(x,b)$$

(iii) $\exists x : \mathbf{Z} | 0 \leq x \leq 5 \bullet x = x * x$

$$\leftrightarrow \exists x : \mathbf{Z} \bullet 0 \leq x \leq 5 \wedge x = x * x$$

演習問題：(i) と (iv) について、同値な論理式を書いてみてください。

解答

$$(i)\ \forall x : \mathbf{Z} \bullet x \in \{1,2,3,4\} \Rightarrow x < 5$$

$$(ii)\ \exists x : Europe \bullet borders(x, albania) \wedge isinEU(x)$$

3.10　述語論理補足

論理式の例 $\mathbf{Q}$ を有理数の集合とする。

$$\forall x : \mathbf{R} \forall y : \mathbf{R} \bullet x < y \Rightarrow \exists z : \mathbf{Q} \bullet x < z \wedge z < y$$

あるいは

$$\forall x : \mathbf{R} \forall y : \mathbf{R} (x < y \Rightarrow \exists z : \mathbf{Q} \bullet x < z \wedge z < y)$$

任意の二つの実数の間には有理数がある。

$$\forall n : \mathbf{Z} | n > 0 \bullet \exists p : \mathbf{Z} \bullet Prime(p) \wedge p \geq n$$

あるいは

$$\forall n : \mathbf{Z} \bullet n > 0 \Rightarrow \exists p : \mathbf{Z} \bullet Prime(p) \wedge p \geq n$$

いくらでも大きい素数がある。

$$\forall x : Mikan | InBox(x) \bullet Delicious(x)$$

$$\forall x : Mikan \bullet InBox(x) \Rightarrow Delicious(x)$$

$Mikan, InBox(x), Delicious(x)$ をそれぞれ「みかんの集合（型）」、「x はこの箱に入っている」、「x はおいしい」を表すとすると ”この箱の中のみかんはみんなおいしいよ！”

$$\exists x : Mikan | InBox(x) \bullet Delicious(x)$$

$$\exists x : Mikan \bullet InBox(x) \wedge Delicious(x)$$

”この箱の中にはおいしいみかんがあるよ ”

$$(\exists x : Mikan \bullet InBox(x) \wedge Large(x)) \wedge (\exists y : Mikan \bullet Inbox(y) \wedge Small(y))$$

”この箱の中には大きいみかんもあるし、小さいみかんもあるよ ”

$$\forall x : Eears \bullet (\exists y : Horse \bullet Belongsto(x, y) \Rightarrow Animalear(x))$$

馬の耳は動物の耳。

$$\forall n : \mathbf{Z} \exists m : \mathbf{Z} \bullet n = 2m \vee n = 2m + 1$$

整数は２の倍数かそれに１を足したものである。（整数は偶数か奇数である）

注 この例のように限量が続く場合には $\bullet$ や () は省いてよい。左から順に読んでいく。もちろん $\bullet$ や () をつけてもよい。

3.11 この章の演習問題

以下演習問題の解答は章末につけます。

なお、「論理とプログラム２００５」の中に演習問題 2003-1,2 などの問題と解答例があるので、練習してみること。

演習問題１ 以下の文章を述語論理の論理式（Z の論理式）で表せ。

1) 首都はどこかの国の都市である。

2) 善人が往生するのだから、まして悪人は往生する。

3) 愛することができる者は幸いなり。

4) 人は誰でも誰かのことを気にかけるものであるが、その人が自分を気にけけるとは限らない。

演習問題２（参考書１、p.29, 2.6)）b) 3 より小さい整数は 7 と等しくない。

c) 9 より小さい偶数は奇数でない。

d) EU に所属して、ベルギーと国境を接する、ヨーロッパの国がある。

演習問題３ 2001 年度の試験問題の一部です。

１ 次の文章を、述語論理の式として、言語 Z の方法で表現せよ。

あるスポーツクラブで、そのすべての会員が同じ県の住人である。

（注：「スポーツクラブ」を表す型を $Sports$、「人」を表す型を $Person$、「県」を表す型を $Prefecture$、「y が x の会員である」を $Member(y, x)$、「y が z の住人である」を $Inhabit(y, z)$ とおく。）

2 次の論理式の型（X, Y）と述語（L）の意味を自分で決め、論理式をできるだけ自然な日本語で述べよ。

$\forall x : X \exists y : Y \bullet L(x, y) \wedge \neg L(y, x)$

3 （参考）　音楽祭の切符販売の管理を表すスキーマ ConcertTicket を書こう。

音楽会が何種類かあり、各チケットは番号と音楽会の種類によって記録される。チケットは一人３枚まで、と決まっている。

管理すべきことは、発行されたチケット、販売されたチケット、残りのチケット、の各集合の相互関係と、販売の際に一人３枚を超えないこと、である。

（注:「番号」､「音楽会」､「聴衆」､ の型をそれぞれ $Number, Concert, Person$ で表す。また､「（発行される）チケットの集合」､「購買者」､「販売されたチケットの集合」､「残りチケットの集合」を表す変数をそれぞれ $tickets, person, issued, left$ で表す。）

本章演習問題の解答例

演習問題１　1) 国の型（国々の集合）を $Country$、首都の型（すべての首都の集合）を $Capital$、「ca は co の都市である」を $isacityof(ca, co)$ とおくと、

$$\forall ca : Capital \exists co : Country \bullet isacityof(ca, co)$$

2) 元は「善人なおもって往生す。いわんや悪人おや」です（と、思う !?）。この論理的意味は「善人が往生する。ゆえに悪人は往生する」であるが、「ゆえに」の由来は「善人が往生するならば悪人は往生する」ということなので、ここではこの最後の文章を論理式で表す。

人々（人類）の型を $Human$、「$person$ が善人である」を $good(person)$、「$person$ が悪人である」を $bad(person)$、「$person$ が往生する」を $heaven(person)$ とおくと、

$$(\forall person : Human \bullet good(person) \Rightarrow heaven(person))$$

$$\Rightarrow (\forall person : Human \bullet bad(person) \Rightarrow heaven(person))$$

注　上の論理式は、二つの論理式を $\Rightarrow$ でつないだものである。それぞれに変数 $person$ が現れるが、これらは独立で互いに影響しない。別の変数を使ってもよい。積分変数のようなものと考えると分かりやすい。

3) 「愛することができる」は「誰かを愛することができる」の意味とする。「p は q を愛することができる」を $love(p, q)$、「p は幸いである」を $happy(p)$ とおくと

$$\forall p : Human \bullet (\exists q : Human \bullet love(p, q)) \Rightarrow happy(p)$$

4) 「p が q のことを気にかける」を $consider(p, q)$ とおくと

$$(\forall p : Human \exists q : Human \bullet consider(p, q))$$

$\wedge\neg(\forall p : Human \forall q : Human \bullet consider(p,q) \Rightarrow consider(q,p))$

演習問題 2

$$\text{b)}\ \ \forall n : \mathbf{Z} | n < 3 \bullet n \neq 7$$

$$\text{c)}\ \ \forall n : \mathbf{Z} | n < 9 \bullet Even(n) \Rightarrow \neg Odd(n)$$

ただし $Even(n), Odd(n)$ はそれぞれ n は偶数である、奇数である、を表すものとする。これらを数学的に表すには、たとえば次のようにすればよい。

$Even(n)$：$\exists m : \mathbf{Z} \bullet n = 2m$

$Odd(n)$： $\exists m : \mathbf{Z} \bullet n = 2m + 1$

$$\text{d)}\ \ \exists co : Europe \bullet InEU(co) \wedge Borders(co, belgium)$$

演習問題 3　1.　$\exists x : Sports \exists z : Prefecture \forall y : Person \bullet Member(y,x) \Rightarrow Inhabit(y,z)$

2.　X を「京都市民」、Y を「大阪市民」、$L(x,y)$ を「x は y のなまえを知っている」とする。

京都市民の誰にでも、ある大阪市民がいて、その京都市民はその大阪市民（のなまえ）を知っているが、その大阪市民はその京都市民を知らない。

3.

$- - - - - - - - - ConcertTicket - - - - - - - - - - -$

$tickets : P(Concert \times Number)$

$issued : P((Concert \times Number) \to Person)$

$left : P(Concert \times Number)$

$- -$

$dom\ issued \cup left = tickets$

$\forall person : Person \bullet \sharp(issued \rhd \{person\}) \leq 3$

4　集合論

「集合と論理」等でみたように、一定の集合の記述方法がある。(情報系数学入門の第一章参照)

集合の記法はいろいろあった。どの場合も集合の要素を中括弧 { } の中に書く。

$\{2,4,6\}$,　$\{2,4,6,\cdots\}$, などは集合の要素を列記するか、要素のいくつかを列記してその先を推測できるようにする、記法である。また、たとえば、$P(x)$ が個体 x のある性質を表すときに

$$S == \{x|P(x)\}$$

とも書く。これは「集合 S は $P(x)$ という性質を満たす個体 x の集まり」ということを表現する。このとき問題になるのは、$P(x)$ がどのように表されるか、また、個体 x はどの範囲で考えるか、ということである。通常では x のとり得る値の範囲の指定は全体集合による。すなわち x が集合 U の要素である、という制限のもとに集合 S を決定することを

$$S == \{x \in U|P(x)\}$$

と書く。性質 $P(x)$ は論理式で表す。

以上で == は左辺は右辺の略記、あるいは名前、であることを示す。言語 Z における用法である。S はつねにその右辺の内容を持つ。「 ⋯ を S とおく」という意味である。

以下では全体集合の代わりに型を使った、Z における集合の記法を学ぶ。

二つの集合 A, B が等しい(全く同じ要素をもつ)ことは、$A = B$ と表す。たとえば、

$$S_1 == \{x \in \mathbf{N}|\exists \in \mathbf{N} \bullet x = 2y\}; S_2 = \{0,2,4,6,\cdots\}$$

ならば、$S_1 = S_2$ である。あるいは、$S_1 = \{0,2,4,6,\cdots\}$ でもある。

4.1　数え上げ

例:

$\{2,3,5,7,11,13,17\}$:最初の 7 個の素数

$$Reprot ::=' Okay'$$

$$|\quad \text{'At top of document'}$$

$$|\quad \text{'At bottom of document'}$$

または

$$Report == \{\quad \text{'Okay', 'At top of document', 'At bottom ofdocument'}\quad\}$$

これらは仕様におけるメッセージの集合（型）の記述に使われる。::= も ==と同様の意味をもつ。

$a..b$ は整数の範囲を示す。すなわち、a から b（a, b を含む）までの数の集合を表す。すなわち、

$$a..b == \{c : \mathbf{Z} | a \leq c \leq b\}$$

たとえば、

$$89..94 = \{89, 90, 91, 92, 93, 94\}$$

（この場合の = は左辺と右辺が等しいことを表す）

$$Potters == \{Ann, David, Beth, Cathy, Elliot\}$$

（Potter 家の人々）

演習問題 3.1　数え上げで表記される集合を自分で作ってみる。

4.2　集合の創造（内包）

集合から別な集合を創造する方法を集合の内包と呼ぶ。

例（Z の記法はごたごたしているので、ここでは「集合と論理」の記法にしたがう）:

$\mathbf{N} == \{n : \mathbf{Z} | n \geq 0\}$　（負でない整数 n の集合 $\mathbf{N}$）

$$\{n : \mathbf{N} | n \neq 0 \wedge n \,\mathbf{mod}\, 2 = 0\}$$

（0 でなく 2 で割り切れる数 n の集合、すなわち、正の偶数の集合）

$n \,\mathbf{mod}\, 2 = 0$ は次の論理式の略記ともみなせる。

$$\exists m : \mathbf{N} \bullet n = 2 * m$$

$$squares == \{n : \mathbf{N} | \exists m : \mathbf{N} \bullet n = m^2\}$$

$$PotterGirls == \{p : Potters | woman(p)\}$$

（Potter 家の女性の集合）

U, V が型 X の部分集合で、$U \subset V$ であるとき、

$$\{x : U | P\} = \{x : V | x \in U \wedge P\}$$

演習問題 5.2　自分で集合を作ってみる。

演習問題　5.3　以下の集合を集合の内包によって定義せよ。それぞれ適当な名前をつける。

1) 1900 年から 2100 年までのすべての閏年の集合。（ちなみに、この集合の個数は？）

2) 3 の倍数である整数（負の数も含めて）の集合。

3) 2) の集合を新しい型としたときに、その中で 4 の倍数である数の集合。

4) 区間 $I = [0, 1]$ で定義された連続関数 $f(x)$ の最大値を与える実数の集合。（実数の型 $\mathbf{R}$ をもつ x で I に属すものの上で $f(x)$ が定義されていることに注意）

5) 4) の関数の零点集合。（$f(x) = 0$ となる x の集合。

5.3 解答　1) $LeapYear == \{n : \mathbf{Z} | 1900 \leq n \leq 2100 \wedge \exists m : \mathbf{Z} \bullet n = 4 * m\}$

2) $ThreeTimes == \{z : \mathbf{Z} | \exists m : \mathbf{Z} \bullet z = 3 * m\}$

3) $ThreeFour == \{n : ThreeTimes | \exists m : \mathbf{Z} \bullet n = 4 * m\}$

4) $Maximum == \{x : \mathbf{R} | 0 \in I \wedge \forall y : \mathbf{R}(y \in I \Rightarrow f(x) \geq f(y))\}$

5) $Zeros == \{x : \mathbf{R} | x \in I \wedge f(x) = 0\}$

演習問題 5.4　論理式の問題 Extra

1. 図書館管理の仕様で、一人当たりの貸し出し数の上限が決まっているとする。（全員に同じ貸し出し可能最大数が決まっている。）この規則を論理式で表現せよ。

2. 上の状況で、立場によって最大貸し出し数が決まっているものとする。この規則を、「各人に最大貸し出し可能数があって」という解釈で、論理式で表現せよ。

演習問題　5.4 の解答　1.　図書館の利用者（登録している人）全体の型を $Reader$、「r さんが m 冊以上借り出せない」を ”r may borrow no more than m books ”と表すことにすると、

$$\exists max : \mathbf{N} \forall r : Reader \bullet r \quad \text{may borrow no more than } max \text{ books}$$

2.　同様に、

$$\forall r : Reader \exists max : \mathbf{N} \bullet r \quad \text{may borrow no more than } max \text{ books}$$

注これらの述語部の書き方はいろいろある。とくに Z には独特の記法がある。ここでは集合の記法に重点をおくので、そこまで立ち入らない。

5　型の生成

ソフトウエア仕様記述の先進技法-Z 言語、ポター他著、田中監訳（プレンティスホール・トッパン）４章より

5.1 基本型の定義

実際に言語 Z を使うときには、Z に組み込まれているいくつかの型（$\mathbf{R}$, $\mathbf{Z}$ 等）の他に、自分で型を宣言することができる。これらは「基本型」（basic type）と呼ばれる。たとえば本と車について語りたいときにはこれらを型として定義しておくと、便利であろう。それには

$$[BOOk, CAR]$$

のように記載する。これを「型定義」と呼ぶ。これによって $BOOK$ と CAR という文字列はこれから作成する仕様においては基本型として使用する、と宣言している。$BOOK$ と CAR の意図された意味は明らかであるが、その内部的な構造は明記しない。それは別途メモをしておけばよい。ただし両者に共通な要素はないことを前提に仕様書を書く。

前出の $Potters$、課題に出た $Human, Flower$ なども自分で定義した基本型である。人と花について語りたいときには

$$[Human, Flower]$$

と型定義をすればよい。

演習問題 6.1 各自基本型の定義をしてみる。

5.2 型合成演算

基本型から型を合成する（合成型）方法（演算）はいろいろある。それらは与えられたいくつかの集合から別な集合を構成する方法と同じである。典型的ないくつかの例を挙げてみよう。

直積（デカルト積） A, B, C などを基本型、あるいはすでに演算で定義された合成型とする。これらから直積演算 $\times$ によって、直積型を合成する：

$$A \times B \quad A \times B \times C \quad A \times A \quad A \times B \times A \quad \mathbf{N} \times \mathbf{R} \times C$$

などなど。同じ型どうしの直積も合成できることに注意。

直積型の意味は、文字通り集合の直積である。したがって集合として考えれば

$$A \times B = \{(a, b) | a \in A, b \in B\}; \mathbf{N} \times A \times B = \{(n, a, b) | n \in \mathbf{N}, a \in A, b \in B\}$$

などとなる。(「集合と論理」では (a, b) を $\langle a, b \rangle$ と書いた。)

$$(a, b, c) : A \times B \times C$$

は、要素（個体）(a, b, c) の型が $A \times B \times C$ であることを表す。この型宣言が示すのは、$a \in A, b \in B, c \in C$ という事実である。

演習問題 6.2 $[X, Y]$（宣言）として、$X == \{a, b, c\}, Y == \{2, 3\}$ とするときに、合成型 $X \times Y$, $\mathbf{N} \times X$ の要素を数え上げのやり方で書いてみる。

解答 $X \times Y = \{(a, 2), (a, 3), (b, 2), (b, 3), (c, 2), (c, 3)\}$

$\mathbf{N} \times X = \{(0, a), (0, b), (0, c), (1, a), (1, b), (1, c), (2, a), (2, b), (2, c), \cdots\}$

べき集合 X が型であるとき、（X から合成される）新しい型として X のべき集合（power set of X）がある。$\mathbf{P}X$ と書く。$\mathbf{P}X$ は、X のすべての部分集合の集まりである。$\mathbf{P}X = \{S | S \subset X\}$。$X$ の要素の個数が n のとき、$\mathbf{P}X$ の要素の個数は 2^n である。X が無限集合でも、$\mathbf{P}X$ の定義は同じであり、"無限の個数"の概念も定義できる。

例 $X == \{1, 2\}$ のとき、

$\mathbf{P}X = \{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$.

$smallnumber == \{1, 2, 3\}$ のとき、

$\mathbf{P}smallnumber = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{2, 3\}, \{3, 1\}, \{1, 2, 3\}\}$

$\mathbf{PN} = \{\emptyset, \mathbf{N}, \{0\}, \{1\}, \cdots, \{0, 0\}, \cdots$,

$\{3, 5, 6\}, \cdots, \{100, 200, 300\}, \cdots, \{2, 4, 6, \cdots\}, \cdots\}$

（各 $i = 0, 1, 2, \cdots$ について i 個の自然数を集めた集合および無数の自然数をいろいろな組み合わせで集めた集合、の集合）

$Players == \{Mary, Elizabeth, Charles\}$ のとき、

$$\mathbf{P}Players = \{\emptyset, \{Mary\}, \{Elizabeth\}, \{Charles\},$$

$$\{Mary, Elizabeth\}, \{Mary, Charles\}, \{Elizabeth, Charles\}, Players\}$$

$[Book]$ とし、$Book$ の要素は本（一冊の本、コピーという）であることを表すとする。このとき $Book$ の部分集合は本の集合である。ハナコの部屋にあるすべての本、産大の図書館の蔵書、などはその例である。

$$library : \mathbf{P}Book$$

（$library$ の型は「ある本の集合」, a set of some copies of books, である）は、$library$ が本の集合であることを示す。つまり図書館とはその蔵書である、と考える。

例 べき集合を使った文章の例

1. 「図書館には図書館司書がいる」（$[Book], [Person]$ を宣言しておく）:

$$\forall library : \mathbf{P}Book \exists p : Person \bullet Librarian(p, library)$$

ただし $librarian(p, library)$ は、p が $library$ の「図書館司書である」ことを表す述語とする。

2. 神山クラブの型を $Koyama$（会員の集合）とし、$[Koyama]$ と型宣言をする。「神山クラブには Hanako と仲良しのグループがある」：

$$\exists G : \mathbf{P} Koyama \forall g : Person \bullet g \in G \Rightarrow GoodTerms(Hanako, g)$$

ただし、$GoodTerms(x, g)$ は「x と g は仲良しである」を表す述語とする。

3. $[Ticket], [KyotoBand]$ と宣言。ただし $KyotoBand$ は「京都バンド」という名のバンド（その楽団員の集合）、$Ticket$ は、そのバンドの６月公演のチケットの集合とする。「m がチケット（番号で区別される）t を売らなければならない」を $MustSell(m, t)$ とするとき、

「京都バンドのメンバーはそれぞれ売らなければならないチケットを割り当てられる」：

$$\forall m : KyotoBand \exists T : \mathbf{P} Ticket \bullet \forall t : Ticket(t \in T \Rightarrow MustSell(m, t))$$

関係の型 $Parent(p, q)$ が ”p は q の親である ”を表すものとする。$Person$ が人々の集合とするとき、$Parent(p, q)$ が成り立つ p と q の対を集めた集合 R_P は Z では

$$R_P == \{p, q : Person | Parent(p, q) \bullet (p, q)\}$$

と書ける。これは Z 以外ではあまり使われない記法なのでここでは省略的に

$$R_P == \{(p, q), p, q : Person | Parent(p, q)\}$$

と書く。正式には

$$R_P == \{x : Person \times Person | \exists p, q : Person \bullet Parent(p, q) \wedge x = (p, q)\}$$

R_P は直積 $Person \times Person$ の部分集合である。R_P が分かれば、$Parent(p, q)?$ も決まる。型でいえば、$R_P : \mathbf{P} Person \times Person$（分かりにくいので $\mathbf{P}(Person \times Person)$ とも書く）。

一般にそれぞれ集合 A, B の要素 p, q について、$H(p, q)$ が p と q の関係を表すとするとき、集合 $R_H == \{(p, q), p : A, q : B | H(p, q)\}$ は $A \times B$ の部分集合であり、

$$(p, q) \in R_H \quad \text{if and only if} \quad H(p, q)$$

となる。逆に、$\mathbf{P}(A \times B)$ の任意の要素 R は

$$H_R(p, q) \quad \text{if and only if} \quad (p, q) \in R$$

によって、関係 H_R を定義できる。したがって H と R_H、R と H_R、を同一する。H あるいは R_H の型は $\mathbf{P}(A \times B)$ である。Z では ”関係 ”を強調する

ために、$A \leftrightarrow B$ という記法を使う。すなわち $A \leftrightarrow B == \mathbf{P}(A \times B)$。型としては、全く同じものである。記法がちがうだけである。

例 上の親子関係の他に例はいろいろある。(「情報系の数学入門」も参照のこと。

1. $Potters == \{Alice, Kate, Beth, Huw, Ben\}$,
$Vehicles == \{Bicycle, Tricycle, Unicycle\}$,

$$Rides == \{(Alice, Bicycle), (Huw, Bicycle), (Ben, Unicycle), (Kate, Tricycle)\}$$

とするとき、$rides(p, x) \leftrightarrow (p, x) \in Rides$ で $rides(p, x)$ という関係は ”p が x に乗る ”を表す。このとき $rides : Potters \leftrightarrow Vehicles$ である。

2. 平面上で、点 (P, Q) が楕円 $2x^2 + 3y^2 = 5$ の内部（周は含まない）にある、という関係 $ellipse$ は $ellipse(P, Q) \leftrightarrow 2P^2 + 2Q^2 < 5$ と表される。$ellipse : \mathbf{R} \leftrightarrow \mathbf{R}$ である。

3. $Players$ は前のとおりとする。

$$Instruments == \{Violin, Piano, Trumpet\}$$

とするとき、

$$concert == \{(M, P), (E, V), (E, P), (C, D)\}$$

とおけば、$concert : \mathbf{P} Players \times Instrument$。

演習問題 6.3 (1)　$[shape, color]$ とし、それぞれ次ぎの集合とする。

$$shape == \{round, square\};\ color == \{red, green, blue\}$$

このとき、$shape \times color$, $color \times shape$ はそれぞれどのような集合か？また、$shape \times color$　の要素のうち、好きな組み合わせを集めたもの $myfavorite$ を書き、その型を示すこと。

(2) $Potters, Instrument$ は前のとおりとし、$[Potters, Instrument]$ とする。Potters 家の人々とそれぞれが演奏する楽器を適当に選んで、Potter 家と楽器を関連付けてください。

解答 (1) $shape \times color = \{(round, red), (round, green),$

$$(round, blue), (square, red), (square, green), (square, blue)\}$$

$$color \times shape = \{(red, round), (green, round), (blue, round),$$

$$(red, square), (green, square), (blue, square)\}$$

$shape \times color$ のうちで、好きな組み合わせの例：

$$myfavorite == \{(round, blue), (square, red)\} : \mathbf{P}(shape \times color)$$

(2) Potter 家の人とその人が演奏する楽器の対の集合の例：

$$Play ==$$

$$\{(Alice, Piano), (Alice, Trumpet), (Kate, Violin), (Huw, Violin), (Ben, Trumpet)\}$$

$Play : \mathbf{P}(Potters \times Instrument)$ であり、$x : Potters, y : Instrument$ とするとき、$x\ plays\ y \equiv (x, y) \in Play$ とすると、$x\ plays\ y$ は ”Potter 家の人々と楽器の関係 ”であり、x さんが楽器 y を演奏する ”を表す。したがって $plays : Potters \leftrightarrow Instruments$ となる。

写像の型 $(x, y) \in S$ であるときに、x と y が S という関係にある、ということができる（上述）。このときに x に y が（S という関係で）対応する、と考えて、x が y に（S によって）対応する、あるいは写像される、とも言う。写像であることを明記するために $x \mapsto y$ と書く。$(x, y) \in S$ は $x \mapsto y \in S$ と書ける。上の例では形と色、人々と乗り物、人々と楽器、の間の対応がある。y を x の写像 S による値という。$S : X \leftrightarrow Y$ のとき、X を写像 S の定義域、Y を値域という。
例：$myfavorite = \{round \mapsto blue, square \mapsto red\}$

$$Play = \{A \mapsto Piano, A \mapsto Trumpet, K \mapsto Violin, H \mapsto Violin, B \mapsto Trumpet\}$$

これらは関係を写像とみなすもので、型は関係のときと変わらない。一つの個体に対して複数の値が対応することがあるので、多価関数とも呼ばれる。

演習 6.4 1. $Potters$ と $Vehicles$ の例の $Rides$ を写像の形式で書き直してみる。

2. Williams 家の子供たちは Ann, Beth, John であり、母親は Mary である。また、Smith 家の子供たちは、Cathy, David であり、母親は Nancy である。両家併せた母親の集合と子供たちの集合をそれぞれ基本型として、それぞれの母親にその子供たちを対応させる写像 Motherhood を記述せよ。

解答 1. イニシャルで略記する。

$$Rides == \{A \mapsto Bi, H \mapsto Bi, B \mapsto U, K \mapsto T\}$$

2. $[Mothers, Children]$、$Mothers == \{Mary, Nancy\}$、$Children == \{Ann, Beth, John, Cathy, David\}$ とおくと、（名前はイニシャルで略記）

$$Motherhood == \{M \mapsto A, M \mapsto B, M \mapsto J, N \mapsto C, N \mapsto D\}$$

$$Motherhood : Mothers \leftrightarrow Children$$

関数の型 写像のなかでとくに値が 1 個のものを関数（一価写像）と呼ぶ。上のほとんどの例は多価なので、この意味の関数ではない。ふつう数学でいう関数はこの意味の関数である。

例 福引祭りがある。引くのは数字の書いてある紙片、当たるのはデジカメ、CD、ティシュー。$X ==$ 4 桁の数字の集合;$Y == \{Degitalcamera, CD, tissue\}$, $[X, Y]$ とする。福引で 4 桁の数字のうち 1234 が当たれば Degitalcamera、下２桁が 45 ならば CD、それ以外はティシューが当たる。このとき、数字と景品の対応は一価である。この関数を $Fukubiki$ とすると、その型は $X \leftrightarrow Y$。

$$Fukubiki == \{1234 \mapsto Dg\} \cup \{ij45 \mapsto CD | 1 \leq i, j \leq 9\}$$

$$\cup \{klij \mapsto tissue | klij : X, klij \neq 1234, ij \neq 45\}$$

関数はそのグラフによって表現できる。$f : A \to B$ のグラフは（f が全域的として）$G_f == \{(x, y), x : A, y : B | y = f(x)\}$ と表記される。G_f がわかれば、$f(x)$ も決まる。G_f は A と B の直積の部分集合である。すなわち $G_f \subset A \times B$、または $G_f \in \mathbf{P}(A \times B)$。型でいえば、$G_f : \mathbf{P}(A \times B)$、あるいは $G_f : A \leftrightarrow B$。$f$ の定義域が $C \subset A$ のとき、略記として

$$G_f = \{(x, f(x)), x : A | x \in C\}$$

と書ける。あるいは $G_f = \{x : A, y : B | x \in C \wedge y = f(x) \bullet (x, y)\}$ とも書ける。

演習問題 6.5 1. $Cities ==$ いくつかの都市の名前、$Countries ==$ いくつかの国の名前、を各自適当に決めて、$[Cities, Countries]$ とする。$IsIn$ を、

都市 ci が国 co にあるとき、ci に co を対応させる関数とする。（一つの都市は一つの国に所属するものとする。歴史的には一つの都市が複数の国に分割されていたこともある、ということはとりあえず無視して）$IsIn$ とその型を記述せよ。

2. $f(x) = \frac{1}{sin x}, x \in \mathbf{R}$ とするとき、f のグラフ G_f を記述せよ。関数の定義域に注意すること。

解答 1. 例：$Cities == \{Tokyo, Kyoto, Chongqing\}$

$$Countries == \{Japan, China, Singapore\}$$

$IsIn == \{Tky \mapsto Jpn, Kyt \mapsto Jpn, Chngqng \mapsto China\} : Cities \leftrightarrow Countries$

2. $G_f == \{(x, \frac{1}{\sin x}), x : \mathbf{R} | \sin x \neq 0\}$, あるいは

$$F_f = \{z : \mathbf{R} \times \mathbf{R} | \exists x : \mathbf{R} \bullet \sin x \neq 0 \wedge z = (x, \frac{1}{\sin x})\}$$

$\sin x \neq 0$ は x についての条件で書けば、$\forall n : \mathbf{Z} x \neq n\pi$ となる。

例題 NIP ホビーショップの在庫品管理仕様の一部では $Stock$ という関係が必要である。すなわち、NIP ホビーショップで扱う各品目の集合を $Item$ とすると、$Stock$ は (i,n)、ただし、$i \in Item, n \in \mathbf{N}$、の対の集合であり、一つの品目についてその個数は一つ決まっている（そのときの在庫数）。

$Stock == \{s : Item \leftrightarrow \mathbf{N} | \forall i \in Item \forall n, m : \mathbf{N} \bullet (i,n) \in s \wedge (i,m) \in s \Rightarrow m = n\}$

この記法では、$Stock$ の要素が型宣言とともに集められている。

記法 $f : A \leftrightarrow B$ が関数であるとき、f の型を $A \rightarrow B$ と書く。

6 論理式の例

○ $\frac{1}{\sin x}$ は x が 0 でないときに定義されている（値がある）。

$$\forall x : \mathbf{R} \bullet x \neq 0 \Rightarrow \exists y : \mathbf{R}(y = \frac{1}{\sin x})$$

○産大の蔵書の一部分は９号館にある。ただし蔵書は本の識別子（ISBN 番号と思えばよい）と産大における個々の番号（同じ本が何冊かある場合の識別子 $1, 2, 3, \cdots$）の対からなる登録番号で表すものとする。本の型を $Book$、「９号館にある」を $IsIn9(x)$ とする。

$$\exists X : \mathbf{P}(Book \times \mathbf{N}) \bullet \forall x : Book \times \mathbf{N}(x \in X \Rightarrow IsIn9(x))$$

○ HP「論理とプログラム２００５」に掲載されている「演習問題 2003 − 1」の解答は、Z の形式でなく、通常の論理式で書いてあります。まず、これら

を読んで理解してください。次に、それを Z の記法で書いてみてください。
たとえば、
（６）愛することができるものは幸いである。
の解答は

$$\forall x(Human(x) \wedge AbletoLove(x) \Rightarrow Fortunate(x))$$

（述語の意味は読んでわかるとおり）Z の記法では、たとえば

$$\forall x : Human \bullet AbletoLove(x) \Rightarrow Fortunate(x)$$

となる。

7　述語論理の論理式について

述語論理の論理式についていくつかの重要な概念がある。

7.1　記号の出現

「情報系の数学入門」の p.134[4] 参照。

論理式の中に現れる記号または記号列は、その論理式の中で、現れる場所も含めて「出現する」という。たとえば、上述の例題

$$\forall x : Human \bullet AbletoLove(x) \Rightarrow Fortunate(x)$$

において、x は 3 箇所現れる。そのそれぞれがこの論理式に出現する、という。現れる場所が違うものについては異なる出現とみなす。$AbletoLove$ はこの論理式に出現する述語記号である。

$\forall x, \exists x$ は x を $\forall, \exists$ でそれぞれ束縛する。すなわち x の働きを制限する。において、$\forall x$ の部分で以下の x の振る舞いを規定する。すなわち、$\bullet$ 以下のことが成り立つかどうか、x に $Human$ のすべての要素を当てはめて調べなければならない。このとき、$\bullet$ 以後の部分は最初の $\forall$ の**有効範囲**にある、といい、変数 x は最初の $\forall$ によって束縛されている、という。実際には $\bullet$ 以後のすべての記号または記号列は最初の $\forall$ によって束縛される。この場合、x のすべての

束縛された変数には代入ができない。いくつかの要素について肯定的な結果ができても、「すべての要素について $\cdots$ が成り立つ」とはいえないからである。

$\exists x$ についても同じことがいえる。