コンピュータ概論 A/B

-- Programming on Mathematica (2) --

数学科 栗野 俊一

伝言

私語は慎むように !!

- 教室に入ったら
  - 直に Note-PC の電源を入れておく
    - Network に接続し、当日の資料に目を通す
    - skype に Login する
    - Windows Update をしておこう
- やる気のある方へ
  - 今日の資料は、すでに上っています
    - どんどん、先に進んでかまいません
- そろそろ試験の準備を始めよう
  - 学んだ内容を確認する (用語/操作方法)
    - 過去の資料を確認 (自分の提出した課題も再利用できないか ?)
    - 過去の全ての課題が *自分だけ* で、*時間内に* 解けるか ?
    - Skype / Google の操作には習熟しているか ?

今後の予定 (本日 〜 試験日)

- 今後の予定
  - 概要
    - 2013/01 : 補講日や、振替で「火曜日」がない
    - コンピュータ概論は今年(2012 年)一杯で終わり
    - 試験 : 試験期間中ではなく、講義期間中 (12/18) に行う
- スケジュール(後ろから..)
  - 2012/12/18 (コンピュータ概論講義最終日)
    - 試験日
  - 2012/12/11 (講義最終前週)
    - 模擬試験日
  - 2012/12/04 (次回:最後の講義)
    - 通常講義 (Mathematica と TeX)
  - 2012/11/27 (本日)
    - 通常講義 (Programming on Mathematica (2))

前回(2012/11/20)の復習

- 講義
  - Mathematica とは : 数式「処理言語」システム
    - 数式を処理するメタ関数を「自分で」「新に」作る事ができる
  - 関数の定義方法
    - 「関数名[引数列]] := 式」
    - cf.「mySquare[x_] := x*x」: x の二乗の計算
  - メタ概念: メタ〜(超〜)
    - システム : 対象(目的語/名詞)と操作(述語/動詞)からなる(システムの理解)
    - 「『システムの操作』を操作する」仕組み
    - 「メタ○○」は、「○○の○○」と置き換える事ができる
    - cf. メタ情報 (情報に関する情報) / メタ数学(数学を対象とする数学)
  - 「言語」とは : 「構文規則」と「意味規則」の対からなる
    - 構文規則 : どのような「形」が言語の要素(文)となるか ? (文型)
    - 意味規則 : その文の意味は何か ? (単語の意味と、構文に対する意味)
    - Mathematica : 構文(シンボル/式, 代入/定義), 意味(値の計算,環境の変更)
- 実習
  - Mathematica の変数の利用法
    - 「=」と「:=」の違い
  - Mathematica の関数の作成方法

本日(2012/11/27)の予定

- 講義
  - Mathematica によるプログラミング基礎
- 実習
  - [演習 1] Mathematica の関数の作成方法
  - [演習 2] 再帰的関数の定義
  - [演習 3] 課題の作成

本日(2012/11/27)の課題

- 今週 (2012/11/27) の課題
  - 次のファイルを提出しなさい
    - 表題 : ペアノの方法による「有理数の差」の関数 qsub を定義しなさい
    - ファイル名 : 20121127-QQQQ.nb (QQQQ は学生番号)
    - 詳しくは、配布した nat.txt の内容を参照
- 先週 (2012/11/20) の課題
  - なし

Mathematica Notebook

- □ Mathematica Notebook (*.nb)
  - ○ Mathematica の計算結果を保存する形式
  - ○ 結果の保存方法
    - ▹ [ファイル]→[別名で保存] : 好きな名前で保存できる
    - ▹ [ファイル]→[保存] (Ctrl-S) : 今の名前で内容を更新する(古い内容は失われる)
  - ○ 結果の利用方法
    - ▹ [ファイル]→[開く] (Ctrl-O) : 以前に保存した内容を読み込む
- □ 電卓としての利用
  - ○「式」を入れて [Shift]+[Enter] (以下、[SE]) すると計算
    - ▹ 入力した式は In[番号] の形で表示される
    - ▹ 計算した結果は Out[番号] の形で表示される
    - ▹ 番号は、[SE] の順番につけられる
  - ○ [SE] の効果
    - ▹「式」の評価が行われる
    - ▹「番号」が増える
    - ▹ In[番号]/Out[番号]が、それぞれ「定義」される

変数への代入と定義(復習)

- 代入と定義
  - 共に変数に値を「割り当て」る
  - 代入 :「変数 = 式」
    - 「式」は「即時評価」される。その評価結果が「変数の値」となる
  - 定義 :「変数 := 式」
    - 「式」は「遅延評価」される。その「式」そのものが「変数の値」となる
  - 変数の値は何度でも変更できる
- 環境
  - 変数名とその値の対応表をもっている
    - 代入も定義も、その対応表を書換えている
    - 書き換える値が違うだけ
- 「式」の評価(変数の場合)
  - 式の中に変数名が含まれていた場合に、それを変数の値に書き換える

関数の利用

- □ Mathematica の関数
  - ○ Mathematica の関数とは
    - ▹「シンボル[引数列]」の形で表現され、値を持つ事ができる「もの」
    - ▹ cf. Sin[Pi] / f[3] / g[x,y^4]
  - ○ 関数の評価
    - ▹「シンボル[引数列]」の形を、その値に置き換える
- □ 引数のパターンマッチ
  - ○「_」(アンダースコア)を利用して「式」の「抽象パターン」が表現できる
  - ○ 例1
    - ▹ next[0] := 1
    - ▹ next[1] := 2
    - ▹ next[_] := 0
  - ○ 例2
    - ▹ fib[1]=1
    - ▹ fib[2]=1
    - ▹ fib[x_]:=fib[x-1]+fib[x-2]

自然数

- ペアノの公理
  - 自然数を定義する公理
    - 0 は自然数である
    - n が自然数なら n+1 も自然数である
    - 上記の二つ以外に自然数はない
- Mathematica でペアノの公理の形の「自然数」を考える
  - 自然数 (これを「ペアノ形式の表現」と呼ぶ事にする)
    - 0 は 0 で表現
    - n + 1 は s[n] で表現
  - 自然数の和
    - padd[0,x_] := x
    - padd[s[x_],y_] := s[padd[x,y]]
  - これを繰り返すと、「分数(有理数)」まで表現できる
    - 「実数」を表現するには、「収束概念」が必要になる

自然数による数の表現 (nat.txt)

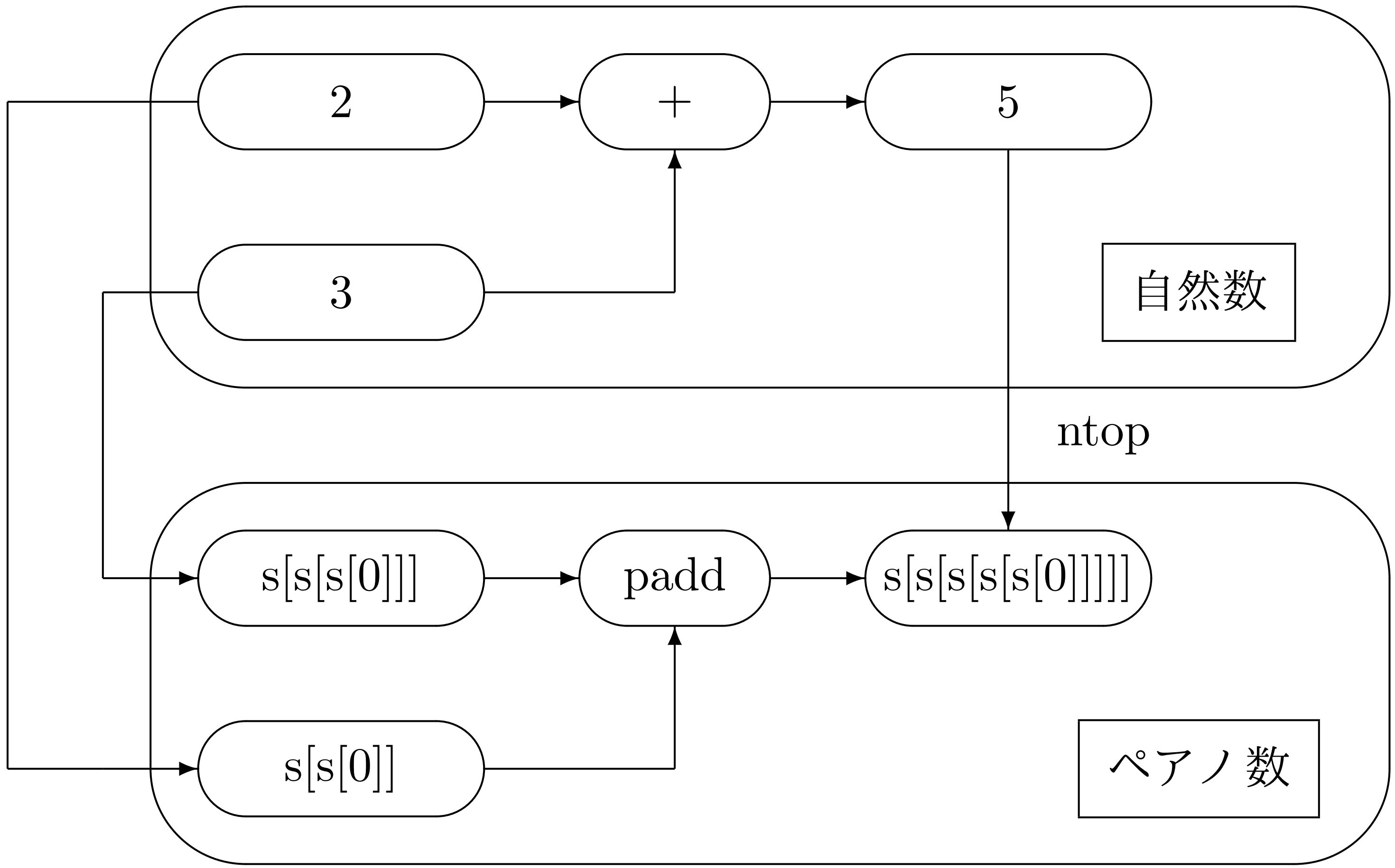
- 自然数
  - 0 と +1 (サクセッサー) のみで作る
    - 3 = 0+1+1+1 = s[s[s[0]]]
- 整数
  - 自然数の二つ組 (m,n) で整数 z を表現する
    - z = m - n => pp[m,n]
    - 同じ整数に対する異なる(m,n)の組があるので、同値関係(〜)を作る
    - 例 pp[4,2] 〜 pp[3,1] 〜 pp[2,0]
- 有理数
  - 整数と自然数の組 (z,p) で整数 q を表現する
    - q = z/p => zp[z,p]
    - 同じ有理数に対する異なる(z,n)の組があるので、同値関係(≡)を作る
    - 例 zp[18,12] ≡ zp[9,4] ≡ zp[6,2]

同型構造

- 空間とは
  - 集合 A と A 上の演算の対 <A,{演算}> [システムと考えてもよい]
    - 例 : 線形空間 -> A:平面ベクトル, A 上の演算: 定数倍やベクトル和
- A, B が同型とは
  - 集合 A, B が演算「・」に関して同型
    - A と B の間に全単射(一対一,上へ)の対応 ϕ (同型対応)があり
    - A 内の演算「・」と B 内の演算「・」が ϕ に対して可換
      - $\phi(a \cdot b) = \phi(a) \cdot \phi(b)$
  - 例: 平面ベクトル (x,y) と複素数(x+yi) : 線形空間として同型
    - 同型対応 : $\phi((x,y)) = x+yi$
    - 演算 (定数倍や和) : $\phi(c(x,y))=\phi((cx,cy))=cx+cyi=c(x+yi)=c\phi((x,y))$
- 同型の利用
  - 同型な場合は、一方の性質を他方の空間で調べる事ができる
    - 例 : 平面ベクトルの性質を複素数で調べる事ができる
    - 例 : 自然数の性質をペアノの形式の数で調べる事ができる

同型

図 1: 可換図



□ 可換：二つの経路が同じ結果になる

「ペアノ形式」と普通の表現の関係

表 1:「S 表現」と普通の表現の関係

概念	普通の表現 (例)	ペアノ形式 (例)	コメント
0	0	0	0 は自然数
自然数 (0 以外)	1,2,3,..	s[0], s[s[0]], s[s[s[0]]], ..	x が自然数なら s[x] も自然数
n の次	n + 1	s[n]	s はサクセッサ (後継) 関数
和	1 + 2	padd[s[0],s[s[0]]]	
差	1 - 2	psub[s[0],s[s[0]]]	引く数の方が大きい場合は 0 にな
積	1 * 2	pmul[s[0],s[s[0]]]	
商	1 / 2	pdiv[s[0],s[s[0]]]	整数割り算 (小数点以下は切り捨て
最大公約数	gcd(1,2)	pgcd[s[0],s[s[0]]]	
整数	1, -1	pp[s[0],0], s[0,s[0]]	自然数の対を同値類で割った物が
整数の正規化		zbar	ペアのどちらか一方を 0 にする
整数の四則	+,-,*,/	zadd,zsub,zmul,zdiv	
有理数	1/2	qq[pp[s[0],0],s[s[0]]]	分子は整数で、分母が自然数の対
有理数の正規化		qbar	約分する
有理数の四則	+,-,*,/	qadd,qsub,qmul,qdiv	

- □ 可換性(well defined)
  - ○ 通常の自然数とペアノ形式の自然数は ntop, pton で同型になっている
  - ○ 個々のペアノ形式の関数は ntop, pton に関して可換になっている
    - ▹ pton[padd[ntop[1],ntop[2]]]=3=1+2