記号論理を J で

SHIMURA Masato
jcd02773@nifty.ne.jp

2012 年 3 月 30 日

目次

はじめに

ライプニッツから始まる論理の記号化は 20 世紀に一応の完成を見た。

数学は抽象化し、記号論理を多用する。情報科学もまた記号論理のユーザーである。

かって *K.E.Iverson* はレオンチェフが持ち込んだレオンチェフ逆行列の計算をサポートする過程でコンピューターで取り扱いやすい数学の研究に入り、*IBM* に移って *10* 余年かけて記号で記述する配列計算言語 *APL* を完成させた。

「日本語から記号論理」を読みながら、*J* 言語で記号論理をどのように取り扱えるかのノートである。

記号論理による自己完結の目論見は、嘘つきのパラドックスという鬼子が暴れだしゲーデルによって止めを刺された。

「私が今言っている事は嘘だ」

1　命題論理と真理表

述語論理では次の論理記号を扱う。

否定記号　…　　$\neg$

結合記号　　　$\vee\wedge\rightarrow$

量化記号　　　$\forall\exists$

命題論理では ∀ ∃ は用いないので簡潔である。

このように僅か数個の記号で論理を展開しょうという遠大な目論見である

1.1　否定記号 ¬

P が論理式であるとき、それを否定した論理式は $\neg P$ である

論理式	J

$\neg P$
 	
$\begin{array}{c\|c} P & \neg P \\ \hline 0 & 1 \\ 1 & 0 \end{array}$
	`not =:  -.`
	
	`   -. 0 1`
	`1 0`
	
	`   not 0 1`
	`1 0`

P の二重否定 $\neg\neg P$ は元に戻り、P と同じである

$$P \Longleftrightarrow \neg\neg P$$

1.2　かつ記号 ∧ とまたは記号 ∨

1.2.1　または ∨

P,Q が論理式のとき、P または Q という意味の論理式 (論理和とも言う)

$$P \vee Q$$

<table>
<tr><th>論理式</th><th>J</th></tr>
<tr><td>$P \vee Q$

<table><tr><th>P</th><th>Q</th><th>$P \vee Q$</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td>
<td><code>or =: +.</code>

<code>0 0 1 1 +. 0 1 0 1</code>
<code>0 1 1 1</code>

<code>0 0 1 1 or 0 1 0 1</code>
<code>0 1 1 1</code></td></tr>
</table>

1.2.2 かつ $\wedge$

P かつ Q と言う意味の論理式 (論理積とも言う)

$$P \wedge Q$$

<table>
<tr><th>論理式</th><th>J</th></tr>
<tr><td>$P \wedge Q$

<table><tr><th>P</th><th>Q</th><th>$P \wedge Q$</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td>
<td><code>and =: *.</code>

<code>0 0 1 1 *. 0 1 0 1</code>
<code>0 0 0 1</code>

<code>0 0 1 1 and 0 1 0 1</code>
<code>0 0 0 1</code></td></tr>
</table>

1.2.3 ド・モルガンの法則

$\wedge$ と $\vee$ 演算は $\neg$ を介してお互いに双対にある

$$\neg[P \wedge Q] \Longleftrightarrow [\neg P] \vee [\neg Q]$$

$$\neg[P \vee Q] \Longleftrightarrow [\neg P] \wedge [\neg Q]$$

1.3 ならば $\longrightarrow$

<table>
<tr><th>論理式</th><th colspan="2">J</th></tr>
<tr><td>$P \longrightarrow Q$

$$\begin{array}{cc|c} P & Q & P \longrightarrow Q \\ \hline 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{array}$$</td><td><code>then=: (+. -.)~</code>

<code>0 0 1 1 then 0 1 0 1</code>
<code>1 1 0 1</code></td><td>$(\neg P) \vee Q$ に同じ

$$\begin{array}{ccc|c} P & Q & \neg P & (\neg P) \vee Q \\ \hline 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{array}$$</td></tr>
</table>

$\longrightarrow$ は多少癖がある。

P が T のとき　–　Q が T なら T、F なら F

P が F のとき　–　Q の如何によらず T

2 全称量化子 $\forall$ と存在量化子 $\exists$

斉藤に「有限集合しか扱わないとすれば量化記号は不要」とある。

2.1 $\epsilon - \delta$ 論法

かの $\epsilon - \delta$ 論法を記号論理で書き下すと次のようになる。

任意に与えられた正の数 ϵ に対し、ある正の数 δ をとると、$| y - x | \langle \delta$ なる全ての y に対して $| f(y) - f(x) | \langle \epsilon$ が成り立つ

$$\neg[\forall\epsilon\exists\delta\forall y[\mid y-x\mid\langle\delta\rightarrow\mid f(y)-f(x)\mid\langle\epsilon]$$

任意の論理式 $P(x)$ に対して

$$\neg[\forall xP(x)]\Longleftrightarrow\exists x[\neg P(x)]$$

$$\neg[\exists xP(x)]\Longleftrightarrow\forall x[\neg P(x)]$$

であるので

$$\neg A(x)\Longleftrightarrow\neg[\forall\epsilon\exists\delta\forall y[\mid y-x\mid\langle\delta\rightarrow\mid f(y)-f(x)\mid\langle\epsilon]]$$

$$\Longleftrightarrow\exists\epsilon[\neg[\exists\delta\forall y[\mid y-x\mid\langle\delta\rightarrow\mid f(y)-f(x)\mid\langle\epsilon]]]$$

$$\Longleftrightarrow\exists\epsilon\forall\delta[\neg[\forall y[\mid y-x\mid\langle\delta\rightarrow\mid f(y)-f(x)\mid\langle\epsilon]]]$$

$$\Longleftrightarrow\exists\epsilon\forall\delta\exists y[\neg[\mid y-x\mid\langle\delta\rightarrow\mid f(y)-f(x)\mid\langle\epsilon]]]$$

$$\neg[P\rightarrow Q]\Longleftrightarrow\neg[\neg P\vee Q]\Longleftrightarrow\neg\neg P\wedge\neg Q\Longleftrightarrow P\wedge\neg Q$$

であるので
P を $\mid y-x\mid\langle\delta$
Q を $\mid f(y)-f(x)\mid\langle\epsilon$
として更に展開すると

$$\neg A(x)\Longleftrightarrow\neg\exists\epsilon\forall\delta\exists y[\mid y-x\mid\langle\delta\wedge\mid f(y)-f(x)\mid\geq\epsilon]$$

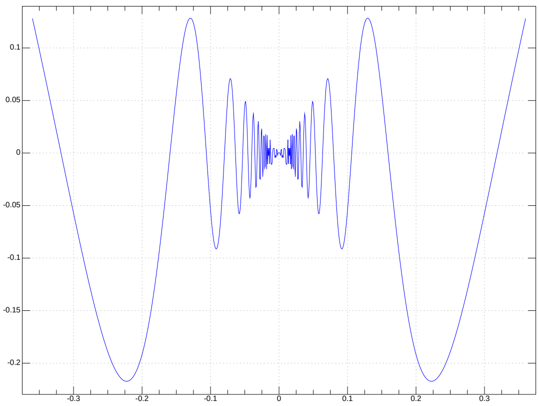
となる。これは次の意味である。
ある ϵ をとると、どんなに小さい δ に対してもある y で

$$\mid y-x\mid\langle\delta\quad\text{かつ}\quad\mid f(y)-f(x)\mid\geq\epsilon$$

となるものが存する。
(証明は専門家にゆだねて) 次の癖のある 2 例で関数が連続か否かをグラフでイメージを掴む。

- *0* ポイントで $\frac{x}{0}$ となるのでこのポイントを外し
- 最後に *expand* で *0* を挿入する
- グラフィックスを描く

$$f_1(x) = \begin{cases} x\sin\dfrac{1}{x} & (x \neq 0) \text{ のとき} \\ 0 & (x = 0) \text{ のとき} \end{cases}$$



連続している

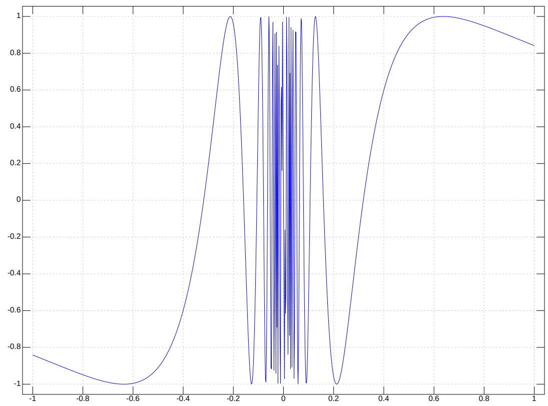
```
test0  steps _0.36 0.36 1000

  pd 'eps d:/mtest.eps'
```

```
test0_sub=: 3 : 0
ind=:-. _ = tmp0=.   % y
tmp1=:(ind # y) *  sin ind # tmp0
tmp2=:ind expand tmp1
)

test0=: 3 : 0
NB. test0_sub steps _1 1 1000
plot y; test0 y
)
```

$$f_2(x) = \begin{cases} \sin\dfrac{1}{x} & (x \neq 0) \text{ のとき} \\ 0 & (x = 0) \text{ のとき} \end{cases}$$



連続でない

```
test1  steps _1 1 1000

  pd 'eps d:/mtest1.eps'
```

```
test1_sub=: 3 : 0
ind=:-. _ = tmp0=.   % y
tmp1=:sin ind # tmp0
tmp2=:ind expand tmp1
)

test1=: 3 : 0
NB. test1 steps _1 1 1000
plot y; test1_sub y
)
```

2.1.1 数列の収束

数列 $\sqrt[n]{n}$ は収束するか

$$\sqrt[n]{n} \langle 1 + \epsilon$$

すなわち

$$(1 + \epsilon) \rangle n$$

を示す

```
2  a=.(i.10), 100 1000 10000 100000 1000000
   a,.(] %: ]) a

     0       0
     1       1
     2 1.41421
     3 1.44225
     4 1.41421
     5 1.37973
     6 1.34801
     7 1.32047
     8 1.29684
     9 1.27652
   100 1.04713
  1000 1.00693
 10000 1.00092
100000 1.00012
   1e6 1.00001
```

3 References

斉藤正彦「日本語から記号論理へ」日本評論社 *2010*

瀬山士郎「初めての現代数学」*(*文庫版*)* 早川書房 *2009*

J 言語の *DL* `http://www.jsoftware.com`

Script の *DL* `http://japla.sakura.ne.jp` の *Workshop* から