アルゴリズム 31－4
(1 9 9 3. 1. 2 5)

純リスプに基づいた頭語的評価関数
- Chaitin のアルゴリズムの欠陥とその修正 -

鎮目浩輔
図書館情報大学

アルゴリズム的情報理論はbinary string *s*の複雑さを、*s*を出力するプログラムの長さとして定義するが、その複雑さにシャノンの情報理論のentorpy に似た性質を持たせるにはプログラムを評価する評価関数*U*に頭語性をもたせることが本質的であることがChaitinにより示されている。本研究ではChaitinが導入したリスプに基づいた頭語的評価関数*U*の定義には欠陥があり、そのため*U*は実際には頭語的になっていないことを示す。さらにその修正法を示す。

The LISP based self-delimiting universal function
- The defect of Chaitin's algorithm and its correction-

Kousuke Shizume
University of Library and Information Science

Abstract

In the algorithmic information theory , the complexity of a binary string *s* is defined as the length of the shortest program which outputs *s* . Chaitin showed that the self-delimiting universal function *U* is essential to formulate algorithmic information theory in a manner closely analogous to the Shannon information theory. We indicate the defect of Chaitin's LISP-based algorithm for *U* that prevents *U* being self-delimiting, and give a corrected algorithm.

1. はじめに

1．1 アルゴリズム的情報理論

アルゴリズム的情報理論（Algorithmic information theory）は，複雑さに対する定量的な尺度を与え，その尺度の性質や応用を研究する理論であり，Solomonoff[1], Kolmogorov[2], Chaitin[3] らによって始められた。この理論はShannonの情報理論[4]と同様に情報の量を定義し、その性質と応用を研究する理論である。ただし0と1からなる記号列、即ち binary string s が与えられたとき、それがもつ情報の量の定義として、Shannonの理論では、大雑把に言ってsの桁数を用いるのに対し、アルゴリズム的情報理論ではsを出力するアルゴリズムの長さ（正確には、もっとも短いものの長さ）を用いる。例えば次のような５０けたの binary string

a = 01010101010101010101010101010101011010101010101010

b = 01011001100110101100010100111010101101001111010111

があったとき、aは"01を２５回プリントする"というアルゴリズムで出力できるのに対し、bの方はそのような簡潔なアルゴリズムでは出力できない。従ってアルゴリズム的情報理論ではaの方がbより少ない情報量を持つと言うことになる。

アルゴリズム的情報理論では、この、アルゴリズムの長さにより定義された情報量が binary string の複雑さ（あるいは乱雑さ）を表わすと考え、complexity（または randomness）と呼んでいる。つまり、ある binary string が与えられ、その0と1からなるパターンを見たときに人間が複雑だと感じるかどうかは、そのパターンを表現するのに長い記述を必要とするか、あるいは短い記述ですむかの問題である、と考え、さらに「表現するのに要する記述の長さ」を「出力するアルゴリズムの長さ」により定量的に捕らえようとしていることになる。

複雑さという概念は日常においてもなじみ深いものであり，それを定量化しようという考えはそれ自体で興味深いと思われる。さらに応用としては、数学基礎論においては不完全性定理の研究に[5]に用いられており、また、物理においてはもともと複雑さや乱雑さという概念は熱力学におけるエントロピーの概念と関係が深いと考えられていたが、その関係をアルゴリズム的情報論を用いて定量的に精密化し、さらにエントロピーの概念を拡張しようという試みが行なわれている[6)~9)]。

1．2 頭語条件

アルゴリズムの長さはそれを表わすプログラムの長さとして定量化される。そのためには、プログラムを解釈し実行する計算機、あるいはその数学的な表現の１つである計算可能な部分評価関数(computable partial universal function、以下では単に部分評価関数と呼ぶ)を準備する必要がある。complexityの性質はこの部分評価関数の選び方に依存するが、Chaitinはこの部分評価関数としてその定義域（部分評価関数をFとしたとき、$F(s)$ が値を持つ binary string s全体の集合）が頭語条件（次章参照）を満たすものを用いれば、complexityがShannonの情報論的entropy に類似した性質を持つことを示した[5),10)]。この性質は応用上重要なものであり、１．１で挙げた応用のそれぞれで本質的な役割を果たしている。この性質を持った部分評価関数Uが実際に存在することを示すために ChaitinはまずTuring machine に基づいたUのアルゴリズムを提案し[10)]、次に一種の純リスプに基づいたアルゴリズムを与えた[5)]。上で述べたようにUはプログラムを実行するための計算機を表わすが、純リスプに基づいたUはリスプの長所

を反映しており簡潔なプログラミングが可能である。アルゴリズム的情報理論においては理論を展開する際に「Uにこのような働きをさせるプログラムが存在するので...」という論法を使うことが多いため、この、簡潔なプログラミングが可能という長所は理論を展開させる上で大きな意味を持つ。

1．3 本論文の目的と構成

しかし、この純リスプに基づいたアルゴリズムには、Chaitinが与えたままでは欠陥があり，Uの定義域が実際には頭語条件を満たさなくなってしまう。本論文ではそのことを示す。さらに、実際に頭語条件を満たすようアルゴリズムの修正を行なう。

本論文の第２章ではアルゴリズム的情報理論におけるcomplexity の厳密な定義と、それに用いる部分評価関数に頭語的条件を課すことの重要性についての紹介を行なう。第３章でChaitinによるUのアルゴリズムについて述べ、第４章においてそれが実際には頭語条件を満たしていないことを示し，さらにその修正法を示す．第５章はまとめである。

2．complexity の定義と頭語条件の重要性[5)]

2．1 complexity の定義

まず、binary string であるが、これは有限個の0と1からなる列である。つまりbinary stringは集合

$$B \equiv \{\Lambda,0,1,00,01,10,11,000,001,....\} \tag{2.1}$$

の要素の１つである。但しΛは空を表わす。また任意のbinary string sの長さを $|s|$ という記号で表わす。さらに、s の頭部になっているbinary stringをsのprefix, 逆にsが頭部になっているものをsのextensionと呼ぶことにする。例えば s が 011であればsのprefixは0と01であり、またextensionは 0110, 0111,01100, ... のそれぞれである。binary string を要素とする集合S（Bの部分集合）が次の条件を満たすときSは頭語条件を満たすという：Sの任意の要素のprefix またはexptension がSに含まれない。
例えば、{0, 10, 111}は頭語条件を満たすのに対し、{0, 10, 101}は満たさない。なぜなら10は101のprefixであるためである。

次に Uを万能評価部分関数(computable universal partial function)で次の１）及び２）の条件を満たすものとする。

１） ２つのbinary string pとqを入力として受け、１つのbinary string を値として出力する。

$$U:\ (p,q)\ \mapsto U(p,q)\quad (\in B) \tag{2.2}$$

但し、partial function であるから入力によっては値を持たない場合がある。
直観的にはpはプログラム、qはデータを表わす。また値$U(p,q)$ はプログラムpに基づきデータqを使って計算を行なった結果であり、値を持たないとは、計算が無限ループにはいるなどして、停止しなくなることである。

2) 任意の binary string qに対し、$U(.,q)$の定義域

$$D(q) \equiv \{p\ :\ U(p,q)\text{が値を持つ}\} \tag{2.3}$$

が頭語条件を満たす。即ち、$s \in D(q)$ならばs のprefix もextension も$D(q)$には含まれない。つまり$U(p, q)$が値を持てば、pの任意のprefixまたはextension r に対して$U(r, q)$は

値を持たないということである。
この2）の条件をUの頭語条件と言うことにする。

このUにより任意の binary string sのcomplexity $H(s)$は次で定義される：

$$H(s) \equiv \min\{|p| : U(p, \Lambda) = s\} \tag{2.4}$$

2．2 頭語条件の重要性

最も重要なことは、Uに対し頭語条件を課すことにより、任意の binary string q に対し、その定義域$D(s)$がKraft 不等式を満たすことである。即ち、Uが頭語条件を満たすならば次が成立する：

$$\sum_{p \in D(q)} 2^{-|p|} \leq 1 \tag{2.5}$$

Chaitin はこの性質を用いて、compleixty の性質で応用上有用なものを証明している。特に重要なのは次の等式である：

$$H(s, t) = H(s) + H(t/s) + 定数 \tag{2.6}$$

但し左辺はs の後にt をつないでできるbinary stringのcompleixtyであり、また$H(t/s)$ は relative complexity

$$H(t/s) \equiv \min\{|p| : U(p, s) = t\} \tag{2.7}$$

である。 この等式はShannon の情報エントロピーが満たすもの[4]と形式的に同じであり、情報論との強い類似性を示唆する。またZurekはこの等式と可逆計算の理論を用いてcomplexityの熱力学への応用を行ない[6]、一方Bennet はKraft不等式を直接用いてcomplexityと熱力学的エントロピーの同等性を論じている[7]。

3．Chaitin によるUの定義

Chaitinは次のI,IIの手順でUを定義し、理論の展開に用いた。

I. 一種の純リスプを定義し、それを用いて２つのbinary string p, qを引数とする部分的評価関数$V(p,q)$を作る。これは頭語条件を満たさない。

II. Vを元にして頭語条件を満たす関数Uを作る。

まずIであるが、

i)最初に一種の純リスプを作る。このリスプは次の特徴を持つ：

- リテラルとして'('と ')' を含む１２８(= 2^7)個の文字をもち、 '('と')'以外の１文字がアトムとして定義される。
- 文字（アトム）の中には、'0' と'1'が含まれる。
- 文字（アトム）の中の１０文字が通常のリスプのCARやEVALに対応する基礎関数に割り当てられる。但しそれらの関数はどのような引数を与えられてもエラーにはならないように定義される。例えば引数の数が多過ぎるときは余計な引き数は無視する。またCARに対応する関数の引数にアトムがきたらそのアトムそのものを値とするなど。
- アトムと'('及び')'を元にしたＳ式の定義やその評価法などは通常のリスプと同じ。

ii)この、１２８個の文字と7桁のbinary stringの間に対応を付ける。（つまり、変換表を用意する。）

iii)任意のbinary string p, qに対し、$V(p,q)$を次で定義する：まず pを左端から7桁づつ変換表に従って文字に換え、文字列を作っていく。この操作を、できた文字列が1つのS式aを構成するまで続ける。次にpの残りにqを継ぎ足し、その各桁を、値が1だったらアトム'1'、0だったらアトム'0'と解釈して、できるアトムの列b_1b_2...をaの引数として与えたS式(ab_1b_2..)をつくる。このS式を純リスプのEVALにより評価し、その値を$V(p,q)$の値とする。なお、pからS式を作れない場合は$V(p,q)$は値を持たないとする。次にIIであるが、Vに頭語条件を付与してそれをUとする。つまり$V(p,q)$が値を持ち、しかもpの任意のprefix またはextension rに対して$V(r, q)$ が値を持たなければ$V(p,q)$の値を$U(p,q)$として定義する。しかし$V(p,q)$と$V(r, q)$ の両方が値を持つ場合でも$U(p,q)$と$U(r, q)$ の両方が値を持つことは禁止しなければならない。こうするためにもっとも簡単なのは、次でUを定義することある。

$$U(p,q) = \begin{cases} V(p,q) & (p\text{のprefix } r \text{ 全てに対し} V(r,q)\text{が値を持たない場合}) \\ \text{値を持たない（上記以外の場合）} & \end{cases}$$

このアルゴリズムが実行できれば確かにUは頭語条件を満たす。しかし「pのprefix 全てに対しVが値を返すかどうか」を調べるということは、prefix のそれぞれに対し停止性検定を行なうことに他ならず、一般的には不可能である。

そこでChaitin はVの計算において行なわれる再帰呼び出しの深さに注目し、次のアルゴリズムを提案した。

1）再帰呼び出しの深さの限度を指定する変数tを用意し、初期値を0にする。

2）$V(p,q)$の評価に必要な再帰呼び出しの深さdepth(p)がt以下かどうかを調べる。

3）pのprefixかextension でt以下の長さをもつものrのそれぞれに対しても$V(r,q)$ の評価に必要な再帰呼び出しの深さdepth(r)がt以下かどうかを調べる。

4）上の2）と3）の結果に応じて次のように条件分岐を行なう

（イ）depth(p)≤tかつ全てのrに対しdepth(r)＞tの場合：$V(p,q)$の値を$U(p,q)$の値とする。

（ロ）depth(p)＞tかつdepth(r)≤tを満たすrが存在した場合：値を返すことを禁止する。

（ハ）depth(p)≤t かつdepth(r)≤ tを満たすrが存在した場合：各rについて$|p|$と$|r|$を比べどのrに対しても $|p|<|r|$ならば$V(p,q)$の値を$U(p,q)$の値とする。さもなくば値を返すことを禁止する。

（ニ）depth(p)＞tかつdepth(r)＞tの場合：tの値を1だけ増やして2）に戻る。

ここで「値を返すことを禁止する」ということは、例えば意図的に無限ループに入るなどして、停止しなくなることを意味する。

Chaitin は上のアルゴリズムを与えただけで、特に説明は与えずに頭語性が保証されることを主張しているが、単にアルゴリズムを見ただけではその意味は分かりづらい。そこで次章ではbinary stringをx-y 平面上の点と対応させた図を考えてアルゴリズムの意味を考える。

4．Chaitin のアルゴリズムの欠陥とその修正

4．1 Chaitin のアルゴリズムの欠陥

Chaitinのアルゴリズムで重要な量はp の長さ $|p|$と、$V(p,q)$の評価に必要な再帰呼び出しの深さdepth(p)である。そこで任意のbinary string s に対し $|s|$ とdepth(s)をそれぞれ x座標、y座標とするx-y 平面上の点を対応させる（右図 1）。例えば長さが3で、再帰呼び出しの深さが2である binary string は図 1 の黒点に対応する。

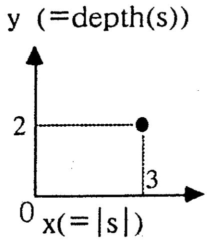

図 1

この図を用いるとChaitinのアルゴリズムは次のような意味を持つことが分かる。つまりステップ 3）においてpのprefixまたはextension rでdepth(r)$\leq t$, $|r|\leq t$を満たすものが存在するかが調べられる。つまり、右図 2 の斜線の領域にprefixまたはextension r が存在するかが調べられることになる。そして存在しなければ、$t<$depth(p)であるかぎりtが増やされ、調べられる領域が広げられる。よって結局、調べる範囲は$|p| \leqq$depth(p)の場合には下図 3 (a)の斜線の領域で、$|p| >$depth(p)の場合には図 3 (b)の斜線の部分になる。そしてこの領域にpのprefixまたはextension r があれば（depth(r)=depth(p)かつ$|r|>|p|$でないかぎり）$U(p,q)$は値をもつことを禁止されることになる。

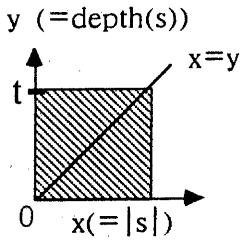

図 2

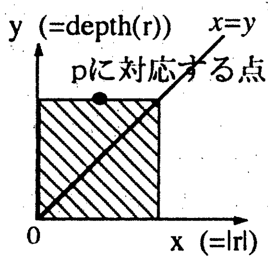

図 3 (a) $|s| \leqq$depth(s)
の場合に調べられる領域

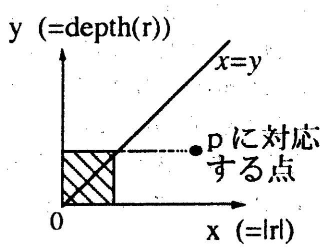

図 3 (b) $|p| >$ depth(p)
の場合に調べられる領域

ではこうすることにより本当に頭語性が保証されるだろうか。否である。実際、$|p| >$depth(p)の場合に、pのprefix r で右図 4 の点R に対応するものがあったとする。するとRが$U(p, q)$ の評価の際に調べられる斜線の領域からはずれているため、$V(r, q)$が値を持つかどうかは$U(p, q)$ の評価に影響しない。しかも$U(r,q)$ の評価の際には調べられる領域は図 4 の、Rを含む実線を一辺とする正方形の内部であるが、そのなかにpに対応する点は入っていない。よって$V(p,q)$が値を持つかどうかが$U(r,q)$の評価に影響することもない。従ってこのアルゴリズムでは$U(p,q)$ と$U(r,q)$の両方が値を返すことを妨げることは保証されない。従って頭

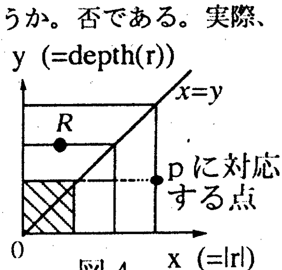

図 4

語条件は保証されなくなる。

4．2 欠陥の修正

この欠陥の修正は容易である。図3を見ればわかるように$|p| > \mathrm{depth}(p)$の場合に調べられる領域を広げて、図4にかかれた、pに対応する点を通る実線を一辺とする正方形の内部になるようにすればよい。いいかえると$U(p,q)$の評価の際に、

$$\{r : |r| \leqq \max\{|p|, \mathrm{depth}(p)\} \text{かつ } \mathrm{depth}(r) \leqq \max\{|p|, \mathrm{depth}(p)\}\}$$

に属するprefix またはextension rがあるかどうかを調べ、あれば値を返すことを禁止し、なければ$V(p,q)$を値として返せばよい。

このようにするためにはChaitinのアルゴリズムにおいて
4）の（イ）を次の（イ'）におきかえればすむ：

（イ'）$\mathrm{depth}(p) \le t$かつ全てのrに対し$\mathrm{depth}(r) > t$の場合：$|p| \leqq t$であったら$V(p,q)$を$U(p,q)$の値として返し、停止する。さもなくばtを1だけ増やし（2）へ戻る。

この修正したアルゴリズムにより、pとそのprefixまたはextension rの両方に対して$V(p,q)$と$V(r,q)$の両方が値を返す場合でも

$$\max\{|p|, \mathrm{depth}(p)\} < \max\{|r|, \mathrm{depth}(r)\}$$

ならば$U(r,q)$は値を返すことを禁止され、逆ならば$U(p,q)$の方が禁止される。さらに

$$\max\{|p|, \mathrm{depth}(p)\} = \max\{|r|, \mathrm{depth}(r)\}$$

ならばpとrのうち長い方（つまり、他方のextensionになっているほう）が、値を返すことを禁止される。
従ってUは確実に頭語条件を満たす。

5．まとめ

本論文においてはChaitinのアルゴリズムではUが頭語条件を満たさないことを指摘し、さらにその修正法を示した。

謝辞

本研究を行なうにあたり、様々な面で援助していただいた増永教授に対し感謝いたします。またChaitinのアルゴリズム及び修正したアルゴリズムを実行するために純リスプをワークステーションへ実装しましたが、その際に有益なアドバイスをいただいた阪口先生に感謝いたします。

引用文献

1) Solomonoff,R.J. A Formal Theory of Inductive Inference. Part I.
Inform. Contr. Vol. 7, p.1-22(1964).
2) Kolmogorov,A.N. Three approaches for defining the concept of information quantity. Inform.Transmission Vol.1, p.3-11(1965)
3)Chaitin,G.J. On the length of programs for computing finite binary sequences. J.ACM. Vol.13, p.547-569(1966)
4) Shannon,W. and Weaver,W. The Mathematical Theory of Communication. Urbana. Univ.of Illinois Press(1949)
5) Chaitin,G.J. Algorithmic information theory.
Cambridge: Cambridge Univ. Press(1987)
6)Zurek,W.H. Algorithmic randomness and physical entropy.
Phys. Rev.A. Vol.40, No.8 p.4731-4751, 1989.
7) Bennet,C. The thermodynamics of computation - a review
Int. J.Theor.Phys. Vol.21 p.905-940(1982)
8)Lloyd,S. and Zurek,W.H. Algorithmic Treatment of the Spin-Echo Effect. J.Stat.Phys. Vol.62 p.819-839(1991)
9)Shizume,K. The decrease of the algorithmic compleity of the spin-echo effect. J. Stat. Phys. issue (1/2)(1993)に掲載予定
10) Chaitin,G.J. A theory of program size formally identical to information theory. J. ACM. Vol.22, p.329-340(1975)