

マルチ集合上での抽象書き換え系を用いた cycle 構造に関する研究

1M-11

鈴木泰博 津本周作 田中博

東京医科歯科大学 難治疾患研究所

{suzuki.com,tsumoto.com}@mri.tmd.ac.jp,
tanaka@cim.tmd.ac.jp

1 はじめに

生物の時間的構造の中で最も特徴的な構造の1つはサイクルである。この構造は細胞質レベルのTCAサイクル、細胞レベルでの細胞周期、多細胞生物のライフサイクルのように生物のあらゆる階層で見られる。従来のmolecular biologyや人工生命等の研究では、サイクル構造の重要性に関する指摘はあったが、この構造の出現、安定化、変遷等の機構の解析は行われてこなかった。今回我々はこれら機構に関し研究するため、抽象書き換え系に基づいた単純な計算モデルを構築し研究を行い、シミュレーションによりかかる簡単な系においても、サイクルの倍周期化、フュージョン、カオス的変遷等の複雑なサイクルの構造が出現することを確認し、その挙動を理論的に特徴付けた。

2 Abstract Rewriting System on Multisets

抽象書き換え系 我々が抽象書き換え系 [1] を用いる理由は、記述の簡便さ、計算系の抽象度が高いこと、従来の研究結果を援用できることの3点である。

定義 1 任意の集合 A 上で定義された書き換えをあらわす2項関係 \rightarrow_a と、その2項関係の和 $R(= \cup \rightarrow_a)$ との対 (A, R) を抽象書き換え系とする。それ以上書き換えることができなくなった記号列を正規型と呼ぶ。

Abstract Rewriting System on Multisets

定義 2 書き換えに用いるマルチ集合を、書き換え集合 (Rewrite set) と呼び S_r と表す。書き換え規則の集合を R と表す。 S_r の空き領域の大きさを C_s と表す。ある書き換え規則 $l \rightarrow r \in R$ が、 $l \in S_r$ の時にもし $(|l| - |r|) \leq C_s$ ならば、 $(S_s - l) \cup (S_s \cup r)$ とする。系には入力が存在し S_r に入力のための空き領域がある限り行われる。

例 $\{aaa\} \rightarrow \{b\}$ を用いて集合 $\{abaa\}$ を書き換える場合、規則の左辺 $\{aaa\}$ は集合に含まれるので、 $\{abaa\}$ は $\{bb\}$ に書き換えることができる。

3 シミュレーション

シミュレーションの条件 C_r を10(ただし初期状態では空集合)、 $abcde$ の5文字を使用、入力は a を規則の試行毎に行った。5000回の書き換えを行なった。書き換え規則としては以下の6つを用いた。

$$aaa \rightarrow b : r_1, \quad ba \rightarrow c : r_2, \quad de \rightarrow a : r_3, \\ c \rightarrow dd : r_4, \quad d \rightarrow e : r_5, \quad d \rightarrow c : r_6.$$

規則の適用は $r_1 \rightarrow r_2 \rightarrow r_3 \dots$ のように順番に行い最後の規則を適用したら最初の規則に戻る (round robin style)。

3.1 シミュレーションの結果

環境の変動が無い場合 規則集合には6規則でのすべての適用順720通りについて系の挙動を調査した。その結果、すべての計算は停止するかサイクルを生じ、計算停止が652回、サイクル発生が68回であった。サイクルはすべて5stepないし6stepであり各々が33回、35回であった。

以下のシミュレーションでは、サイクルが発生する適用順 O_{r_1} として、 $\{r_1 \rightarrow r_2 \rightarrow r_3 \rightarrow r_4 \rightarrow r_5 \rightarrow r_6\}$ を、正規型に至る適用順 O_{r_2} に $\{r_3 \rightarrow r_1 \rightarrow r_2 \rightarrow r_4 \rightarrow r_5 \rightarrow r_6\}$ を用いた。

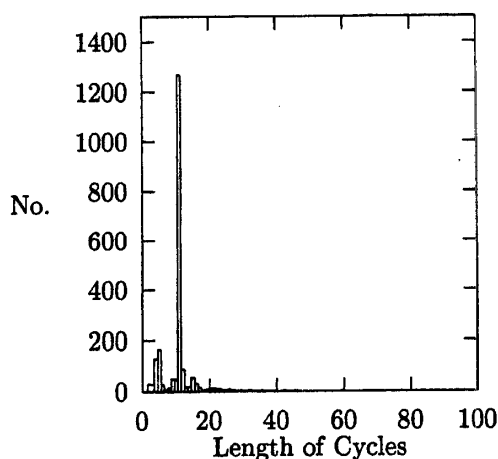
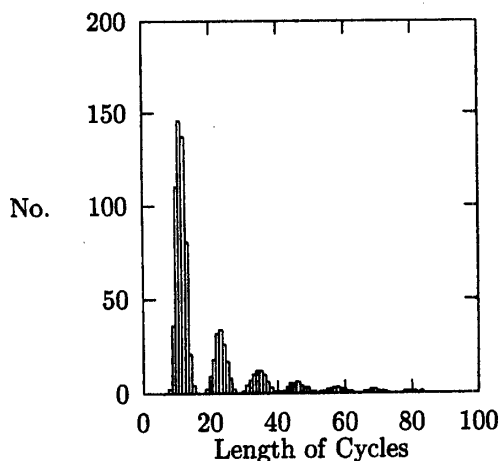
そして、系の挙動が環境の変化にどのように適応するか調査した。環境の変化は、入力の頻度を下げる、適用順に乱雑さを導入することにより実現した。

乱雑さの導入のため以下の特性関数を導入した。 k は0から10までのnegative integerである。

$x \in [0, 1)$ の一様乱数とする。

$$h_\delta(x) = \begin{cases} 1 & (x < \delta) \\ 0 & (x \leq \delta) \end{cases}$$

この関数を用い環境の変化を、各書き換えstep毎にあった入力を10stepsのうち a stepだけ停止する(入力の変動)、固定していた規則適用順を10stepsのうち a stepsだけランダムに規則を選択する(適用順の変動)として実現した。($a = 1.2.3 \dots$)

Figure 1: An Illustrative Example in O_{r1} Figure 2: An Illustrative Example in O_{r2} : Frequency of Input: 0.4, with Rule Order Fixed.

環境を変動させた場合のシミュレーション結果我々は、規則適用順を固定し、入力頻度を減少させる。毎stepごと入力を行い、規則適用順に乱雑さを導入する。規則適用順に乱雑さを導入すると共に入力頻度も減少させる。の3つの場合についてシミュレーションを行なった。

O_{r1} (サイクルが出現する適用順) 適用順を変化させることには弱く、適用順のランダムネスの増加にしたがいサイクル数は単調減少する。適用順を固定し入力頻度を減少させていった場合も同様の傾向がみられた(図1)。

O_{r2} (サイクルが出現しない適用順) 10回に1回ランダムに規則を適用しただけでサイクルが出現した。サイクルの周期は3が最低で数が最も多い。だがその倍周期は存在せず、長周期になるにしたがってその数は単調減少してゆく(図1)。

規則適用順を固定、入力頻度減少の場合、入力を10回のうち9回に減少させると周期5~6のサ

Table 1: Relation between Frequency of Inputs (δ) and Termination of Calculus

δ	$0.7 \geq$	0.6	0.5	0.4	≤ 0.3
O_{R1}	0	0	0	0	0
O_{R2}	100	100	98	13	0

Rule Order: Fixed

イクルとその倍周期も出現する。頻度の減少と共に倍周期も減少するが、頻度10回に4回で突然、多様な周期をもつ長周期のサイクルが数多く出現するのが確認できた(図2)。この場合について詳しく挙動を解析してみるといくつかのサイクルがフュージョンしていることが判明した。

計算の停止性 入力頻度を減少させた場合、 O_{r2} でも10回に5回以下になると計算は停止しない。だが、規則適用順を変動させた場合はたとえ10回に1回でもランダムにすると O_{r1} , O_{r2} いずれの場合も計算は停止する(表1)。

計算の停止性には計算の用いる文字の数と入力、生成される文字の数のバランスが重要であり、その均衡点が10以上ならば計算は停止する。この一般化より次の定理が得られる。定理の成立にはいくつかの概念が必要だがここでは省く。[2]を参照されたい。

定理 1 規則の総数が n 、入力された文字総数を m とする。また、書き換え規則左辺の記号数を l 、右辺を r であらわす。このとき、

$$\sum_{k=1}^n |r_k| + m - \sum_{k=1}^n |l_k| \neq 0$$

ならば ARMS は停止する。

4 まとめ

抽象書き換え系に基づいた単純な計算モデルを構築し、シミュレーションによりこのような簡単な系でも、サイクルの倍周期化、フュージョン等の複雑なサイクルの構造が出現することを確認した。また、計算の停止性につき理論的な性質付けを行なった。

References

- [1] J. W. Klop, Term Rewriting System, Handbook of Logic in Computer Science, 3-62, Clarendon Press, 1992.
- [2] S. Yasuhiro, S. Tsumoto and H. Tanaka, Analysis of Cycles in Symbolic Chemical System based on Abstract Rewriting System on Multisets, Proceedings of the A Life V, MIT press, in printing.