記　号　処　理 71－5
（1993．11．19）

プロダクション規則と局所評価関数にもとづく計算モデル CCM による各種のソート法*

金田　泰**
新情報処理開発機構 (RWCP) つくば研究センタ

著者は，局所情報だけで計算されるプロダクション規則と評価関数とを要素とする確率的な計算モデル「化学的キャスティング・モデル (CCM)」とそれにもとづく計算言語 SOOC を提案している．このモデルは仕様も明確にかきくだせない開放系の問題に適用するために開発したものだが，古典的な問題への適用実験も重要だとかんがえている．そこでこの報告では CCM をソートに適用し，あたらしいソート法をしめすとともに，従来の挿入ソート，交換ソートなどにもとづくソート法をしめす．また，あわせてそれらの実験結果をしめす．この方法では，唯一のプロダクション規則と唯一の評価関数をあたえるだけで，局所情報の参照だけにもとづいてソートをおこなうことができる．また，局所情報だけをつかうことによって生じる興味ぶかい現象についても言及する．

Various Sorting Methods by *CCM*, a Computation Model based on Production Rules and Local Evaluation Functions†

Yasusi Kanada**
Tsukuba Research Center, Real-World Computing Partnership

The author has proposed the Chemical Casting Model (CCM), which is a stochastic computation model whose elements are production rules and evaluation functions that are computed only using local information, and has proposed a computation language SOOC based on CCM. This model is targeted to apply to problems on open systems in which even specifications cannot be clearly written down. However, it is also important to make experiment on applying this model to conventional problems. Thus, CCM is applied to sorting in this paper, and a new sorting method is presented and several sorting methods based on conventional methods, i.e., insertion sort and exchange sort, are presented. The results of experiments are also shown. By these methods, sorting can be done only using one production rule and one evaluation function both of which only refer to local information. Several interesting phenomena caused by computation only using local information are also mentioned.

* この研究の一部は著者が日立製作所中央研究所においておこなったものである．
† Part of this research was done at Central Research Laboratory, Hitachi Ltd.
** E-mail address: kanada@trc.rwcp.or.jp

1. はじめに

金田 [Kan 92a, Kan 94] は，開放系 (環境に対してひらかれたシステム) における仕様も明確でない現実世界の問題をとくための自己組織的計算をめざして，化学的キャスティング・モデルという計算モデルを提案している[注1]．CCM はプロダクション・システムにもとづいているが，従来のそれとはちがって局所秩序度という一種の評価関数と非決定性 (確率的制御) をとりいれている．局所的な情報だけで動作するプロダクション規則と局所秩序度からなる単純かつ汎用的な“プログラム”により，自己組織的に“大域的秩序”をひきだすことをめざしている．

CCM はあたらしい計算モデルなので，それをつかって各種の問題をといてみることが重要である．CCM がその真価を発揮するのは環境に対して開放されたあたらしい型の問題だとかんがえられるが，現在まではその前段階として古典的なとじた制約充足問題，最適化問題や多項式時間でとける問題への適用をこころみている．制約充足問題としては N クウィーン問題 [Kan 92a] とグラフ彩色問題 [Kan 92b]，最適化問題としては巡回セールスマン問題 [Kan 93a] と整数計画問題 [Kan 93d] についてすでに報告した．

この報告では CCM によるソートについてのべる．このばあいも，アルゴリズムを記述せず，できるだけ単純な“プログラム”によって局所的な情報だけにもとづいてソートを実現することをめざしている．CCM によるソートに実用的な価値はないであろうが，この記述実験はより複雑な問題をとくうえでやくにたつであろう．第 2 章ではかんたんに CCM について説明し，第 3 ～ 5 章では CCM にもとづく各種のソート法についてのべる．第 6 章では，いまのところ CCM によって記述できない選択ソートや各種の計算時間が $O(N \log N)$ のソートについてのべる．

2. 計算モデル CCM

この章では化学的キャスティング・モデル (Chemical Casting Model, CCM) について説明する．CCM は化学反応系とのアナロジにもとづく計算モデルであり [Kan 92a]，不完全な情報や非決定的な計算計画のもとでも動作することをめざしている．

CCM の構成要素についてかんたんに説明する．CCM はプロダクション・システムにもとづくモデルである． OPS5 [For 81] などと同様に，データは**作業記憶**におかれる．従来のプロダクション・システムにおける規則ベースすなわちプログラムに相当するものを**キャスタ**とよぶ．CCM は不完全な計画にもとづく計算のためのモデルなので，完全な計画を意味するプログラムということばを使用しない．いまのところ，キャスタはユーザによって記述され，そのままのかたちでつかわれる．

作業記憶にふくまれるべきオブジェクトあるいはデータとしては，つぎのようなものがある (図 1 参照)．**原子**はデータの単位であり，内部状態をもつ．原子にはデータ型があり，それを**元素**ともよぶ．原子どうしを**リンク**によって結合することができ，結合された全体を**分子**とよぶ．リンクは無向でも有向でもよい．無向のリンクは化学結合に似ているが，化学結合には有向のリンクに相当するものはない．また，リンクにはラベルをつけることもできる．

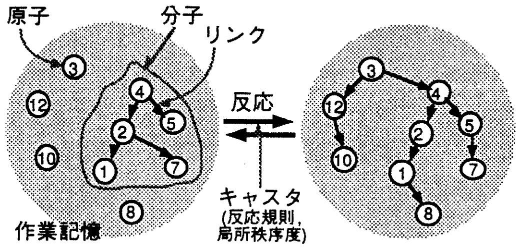


図 1　化学的キャスティング・モデルの構成要素

キャスタは反応規則と局所秩序度とで構成される．**反応規則**はシステムの局所的な (ミクロな) 変化のしかたをきめる規則であり，ユーザにより定義される．ここで「局所的」ということばは，その反応規則によって参照される原子数がすくないということを意味する．反応規則は前向き推論によるプロダクション規則として記述される．したがって，つぎのようなかたちをしている：LHS → RHS.

反応規則は化学反応式に相当するものだといえる．後述する N クウィーン問題やグラフ彩色問題 [Kan 92b] などをはじめとするおおくの単純なシステムにおいては反応規則は 1 個だけ存在するが，複数の変化のしかたをみとめるより複雑なシステムにおいては複数個の反応規則が存在する．

局所秩序度は局所的な“組織化”あるいは“秩序化”の程度をあらわす一種の評価関数であり，作業記憶の局所的な状態が“のぞましい”ほどおおきな値をとるように，ユーザにより定義される．局所秩序度の存在は，通常のプロダクション・システムにくらべたときの CCM のもっともおおきな特徴である．局所秩序度はつぎの 2 つのうちのいずれかのかたちで定義される．(1) **自己秩序度** $o(e)$：1 個の原子 e に対して定義される．(2) **相互秩序度** $o(e1, e2)$：2 個の原子からなる対 $<e1, e2>$ に対して定義される．

[注1] このモデルは金田 [Kan 92a] においては「化学的プログラミング・モデル」とよばれていた．

後述のキャスタのなかには相互秩序度を使用するものもあるが，ここではかんたんのため自己秩序度だけをかんがえる．自己秩序度は規則の適用時に原子ごとに計算されるが，その値は当該原子の内部状態だけでなく，それを始点とするリンクの終点である原子の状態にも依存しうる．

反応はつぎの2つの条件をみたすときにおこる．反応規則の左辺 LHS および右辺 RHS には原子とマッチする1個または複数個のパタンがあらわれるが，第1の条件は左辺にあらわれるすべてのパタンのそれぞれにマッチする原子が存在することである．

反応がおこるとこれらの原子は消滅して，そのかわりに右辺にあらわれる原子が生成される．ただし，左辺と右辺とに対応する原子があらわれるばあいは，その原子は生成・消滅するかわりにかきかえられる．このような規則とそれにあらわれる (左辺および右辺の) パタンにマッチするすべての原子との組をインスタンスとよぶ．ひとつのインスタンスがふくむ原子のうち，反応前に存在するものすなわち左辺にあらわれるものの局所秩序度の総和を "反応前のインスタンス秩序度"，反応後に存在するものすなわち右辺にあらわれるものの局所秩序度の総和を "反応後のインスタンス秩序度" とよぶ．反応後のインスタンス秩序度をあらかじめ計算したものが反応前のインスタンス秩序度よりおおきいとき，すなわち反応によって局所秩序度の和が増加する時だけ反応がおこるというのが第2の条件である．そして，いずれかのインスタンスについて上記の2条件がみたされているかぎり，反応はくりかえしおこる．

ただし，一般には上記の2つの条件をみたすインスタンスは複数個存在する．条件をみたすインスタンスが複数個生成される原因としては，ひとつの規則の条件部をみたす原子の組が複数個存在するばあいと，複数の規則についてその条件部をみたす原子の組が存在するばあいとがある．いずれのばあいでも，これらのインスタンスのうちのいずれがどのような順序で，あるいは並列に反応するかは非決定的であり，反応の順序はシステムが自発的にきめる．反応の順序によらずにのぞむ計算をおこなわせるはたらきをする (すべき) のが局所秩序度である．

上記のような自発性あるいは非決定性を CCM にあたえているひとつの理由は，非決定性のないアルゴリズミックな計算においては，プログラマがあたえた "よけいな制御" によってプログラムの動作が制約され，自己組織的な計算や並列度のたかい計算を不可能にしているばあいがあるとかんがえられるからである [Kan 92b].

しかし，反応の順序をある程度はユーザが制御することができないと，のぞんだ計算を実現できないばあいがある．ユーザはスケジュリング戦略 [Kan 92a] を指定してインスタンスの選択順序を制御し，反応の順序を部分的に制御することができる．スケジュリング戦略は従来のプロダクション・システムにおける競合解消戦略に対応するが，基本的なスケジュリング戦略は，インスタンスをランダムに選択する**ランダム戦略**である．このような戦略は従来はもちいられることがなかった．このほかに，インスタンスを系統的に選択する**系統的戦略**もある．

3. 挿入ソート

この章では CCM にもとづく挿入ソート (insertion sorting [Knu 73]) についてのべる．まず，使用するデータ構造と初期状態・終状態について説明する．

CCM によってソートをおこなうには，いくつかのデータ構造がつかえる．ここでは個々のデータを原子として表現し，それらを昇順または降順にリンクでつなぐことによってソートを実現する．この方法では，たとえば図2の左辺のような初期状態があたえられるとその右辺のような終状態がもとめられる．

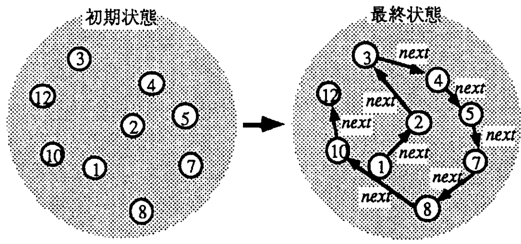


図2　リンクによるソート

図2では，初期状態においてすべてのデータはリンクされていない．しかし，データがリンクによって任意に接続されている状態を初期状態とすることもできる．このばあい，図2の終状態におけるように，すべてのデータが一列にリンクされているとはかぎらない．すなわち，複数の原子からのリンクが同一の原子をさしているばあいもあるし，どの原子からもさされていない原子や，どの原子もさしていない原子が存在するばあいもある．ただし，各原子を始点とするリンクは1個または0個にかぎる．

以下挿入ソートの反応規則と局所秩序度の定義をしめす．これらを実行させるためには SOOC という計算言語によって記述するが，ここではわかりやすくするために，それとはことなる表現をつかう．

図3に挿入ソートの唯一の反応規則をしめす．この規則は，つぎのような条件がみたされたときに適用される．3個の原子 d1, d2, dn (これらのデータの

型はすべて“data”)が存在して，d1 から d2 への *next* というラベルをもつリンクが存在し，dn に関しては *next* というラベルをもつリンクが dn をさしていないし，dn から他のデータへのリンクもない．また，さらに後述の局所秩序度をつかって計算されるインスタンス秩序度に関する条件もあわせてなりたつ．

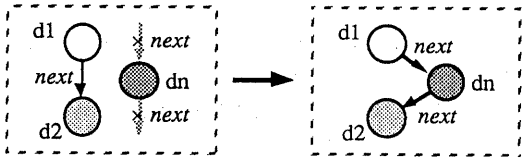


図3　挿入ソートの反応規則

上記の条件がなりたったときには，つぎのようなかきかえがおこなわれる．d1 からのリンクは dn を終点とするようにかきかえられる．また，dn から d2 へのあらたなリンクがつくられる．ここで，左辺において dn を終点とする原子がないという条件が必要なのは，この条件がなければかきかえによって複数の原子が dn を終点とするようになり，データが単鎖をなさなくなるからである．

つぎに，挿入ソートのための局所秩序度の定義をしめす．すなわち，data 型に関する自己秩序度をつぎのように定義する．

$$o_{\text{data}}(d) = \begin{cases} 0 & \textbf{when } d.next = \textbf{nil} \\ 0 & \textbf{when } d.value < d.next.value \\ 1 & \textbf{when } d.value \geq d.next.value \end{cases}$$

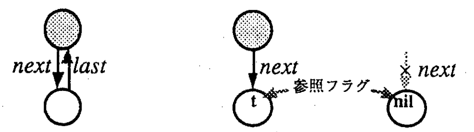
ここで *d.value* はデータ *d* の値，*d.next* はデータ *d* を始点とするリンク，*d.next.value* はそのリンクの終点であるデータの値を意味する．

金田ら [Kan 93b] は CCM にもとづく制約充足問題の解法をしめしている．ソートも制約充足問題の一種とかんがえることができるから，この解法をあてはめることができる．この解法においては局所秩序度は局所的な制約がみたされるとき 1，そうでないとき 0 という値をとるように定義するが，上記の局所秩序度は実際そのように定義されている．なお，上記の局所秩序度は降順のソートをおこなうためのものだが，不等号のむきを逆にすれば同一の規則をつかって昇順のソートをおこなうこともできる．

初期状態としては図 2 の左にしめしたリンクされていないデータを使用するが，上記の規則を適用するためには，初期状態においてもリンクされたデータがすくなくとも 1 組は必要である．そこで，“最大の数”と“最小の数”をあらわす 2 個のダミー・データをリンクしたものを，これらとともに作業記憶にいれておく．この「ひも」にデータを挿入していくことによりソートがすすむ．また，ほかにはいっさいリンクが存在しないことを仮定する[注2]．

ところで，このキャスタを SOOC-93 によって実現するためには，もうひとくふう必要である．すなわち，SOOC-93 においてはリンクを逆方向にたどることができないため，dn を終点とする原子がないことをしらべるときに双方向のリンクを使用するか，または単方向リンクを使用するばあいには参照フラグ (リンクによってさされていないことをしめすフラグ) を併用する必要がある (図 4 参照)．

平均計算時間などを図 5 に例示する[注3]．これは，ランダムに生成した整数データをランダム戦略によってソートした結果である．スケジュリングの変化に対して性能は安定している[注4]．系統的戦略の一種であるふかさ優先戦略でもおなじ傾向をしめす．マッチング回数 (Tests) と計算時間の平均はほぼ $O(N^2)$ であり，平均反応回数は $O(N)$ である[注5]．ただし確率的方法を使用しているため，これらに上限はない．



(1) 双方向リンクの使用　　(2) 参照フラグの使用

図4　SOOC-93 による挿入ソートの方法

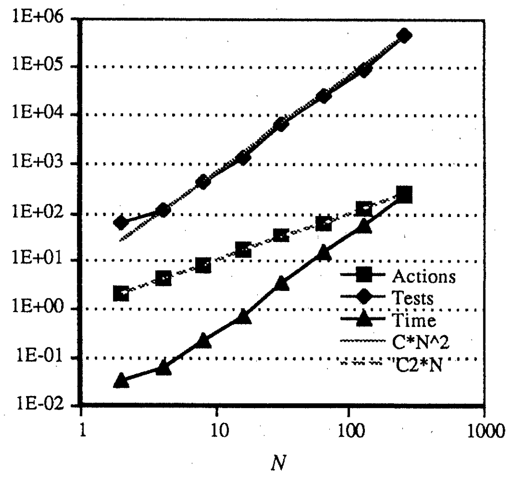


図5　挿入ソートの計算時間 (ランダム戦略)

[注2] 後述する連結ソートのばあいとはちがって，挿入ソートにおいてはこの仮定がないとただしくソートされないばあいがある (第 5 章参照)．

[注3] SOOC-93 の処理系が未完成であるため，構文などがややことなる SOOC-92 の処理系によって測定をおこなった．

[注4] この性質を“スケジュリング感度がひくい”という．

[注5] 測定は Macintosh SE30 上の Macintosh Common Lisp によっておこなった．第 4 章以降でしめす交換ソートや連結ソートの測定結果も同様である．

なお，前記のキャスタを使用すれば，いったんソートが終了したあとでも，データを追加すればそれがソート列にとりこまれる．すなわち，漸進的(incremental) なソートをおこなうことが可能である．

4. 交換ソート

この章では CCM にもとづく交換ソート (exchange sorting [Knu 73]) についてのべる．挿入ソートのばあいと同様にリンクをつかったデータ表現をあつかうこともできるが，それでは規則が繁雑になるため，ここではインデクスをつかってデータの順序を表現する．すなわち，各原子はデータの値とともにインデクスの値をもつ．SOOC-93 は配列というデータ構造をもたないので，このように各原子が陽にインデクスをもつようなデータ構造をつかう．図 6 の左にしめすように，初期状態においてはデータの順序はランダムである．ただし，インデクスの値が一意であることは保証されているとする．実行終了後には，図 6 の右にしめすように昇順または降順にインデクスづけされる (図6 を図2 とみくらべるとよい)．

以下交換ソートの反応規則と局所秩序度の定義をしめす．まず図 7 に唯一の反応規則をしめす．この規則は，インスタンス秩序度に関する条件がみたされる任意の 2 個の原子に作用して，そのインデクス I1, I2 の値を交換する．これらのデータの型は data とする．なお，手続き型言語で記述するばあいには交換するデータ選択のしかたによってバブル・ソートをはじめとするさまざまな交換ソート法がくべつされるが，図 7 の規則においてはデータ選択法が指定されていないので，これらはくべつされない．

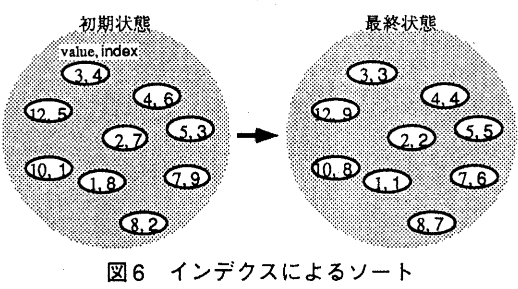


図6　インデクスによるソート

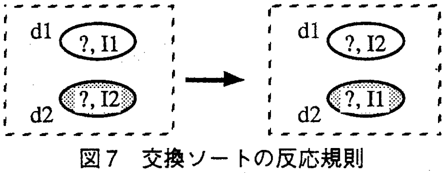


図7　交換ソートの反応規則

つぎに，交換ソートのための局所秩序度の定義をしめす．すなわち，data 型に関する相互秩序度をつぎのように定義する[注6]．

$o_{\mathrm{data}}(d1, d2) =$
　1　**when** $d1.value = d2.value$ **or**
　　　$(d1.value < d2.value$ **and** $d1.index < d2.index)$ **or**
　　　$(d1.value > d2.value$ **and** $d1.index > d2.index)$
　0　**otherwise**

この局所秩序度も金田ら [Kan 93b] における局所秩序度の定義にしたがっていて，0 または 1 という値をとる．また，上記の定義においてインデクスの比較における不等号のむきを両方とも逆にすれば，挿入ソートのばあいと同様に，同一の規則をつかって昇順のソートをおこなうことができる．

平均計算時間などの例を図 8 にしめす．これはランダムに生成した整数データをランダム戦略によってソートした例である．ふかさ優先戦略のばあいもほぼ同様の結果がえられる．マッチング回数 (Tests) と計算時間は $O(N^{\alpha})$ $(2 < \alpha < 3)$ であり，手続き型言語によって記述するばあいよりややよけいに時間がかかっている[注7]．また，反応回数はほぼ $O(N \log N)$ であることがわかる．

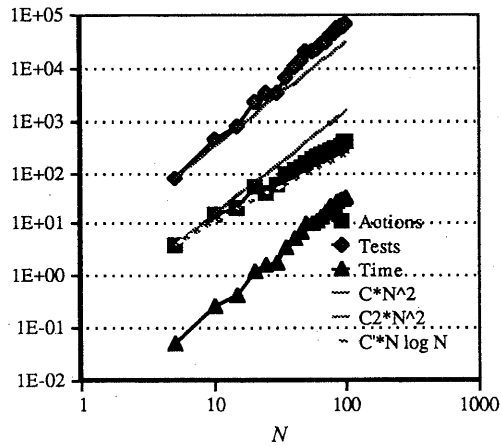


図 8　交換ソートの計算時間 (ランダム戦略)

5. 連結ソート

この章では CCM 特有のソート法についてのべる．

[注6] 反応規則と相互秩序度の対称性を確保するために相互秩序度の定義が複雑になっている．相互秩序度の値が引数の交換したときに変化してもかまわなければ，つぎのようなより単純な定義をあたえることができる．

$o_{\mathrm{data}}(d1, d2) =$
　if $d1.index < d2.index$ **and** $d1.value \le d2.value$ **then** 1
　else 0

[注7] スケジュリングの実装を改善すれば，計算時間を $O(N^2)$ にすることができると予想している．

このソート法を連結ソートとよぶ．従来のソート法のなかには対応するものがないとかんがえられるのであらたな名称をあたえた．ここでしめす連結ソートにおいては，挿入ソートでつかったのとおなじようにリンクによってデータの順序を表現する．

図9に連結ソートの唯一の反応規則をしめす．この規則は，インスタンス秩序度に関する条件がみたされれば任意の2個の原子 d1, d2 に適用される．すなわち，d1 を始点とするリンクはあってもなくてもよいし，d2 を終点とするリンクもあってもなくてもよい．これらのデータの型を data とする．条件がみたされれば，d1 を d2 に連結する．すなわち，もし d1 を始点とするリンクがあらかじめ存在したばあいには，それが d2 を終点とするようにかきかえられるし，d1 を始点とするリンクが存在しなかったばあいには，あらたな d2 へのリンクが生成される．

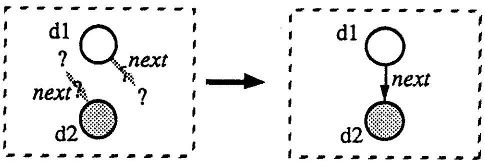


図9　連結ソートの反応規則

つぎに，連結ソートのための局所秩序度の定義をしめす．data 型に関する自己秩序度として定義する．

$$o_{\mathrm{data}}(d) = \begin{cases} 0 & \textbf{when } d.next = \textbf{nil} \\ 1 / (d.value - d.next.value) & \textbf{when } d.next \neq \textbf{nil} \end{cases}$$

ここで，作業記憶には同一の値をもつデータが存在しないことを仮定する．そうすれば，上記の式の分母は0にはならない．この条件については，あとでふたたびふれる．

図9の規則は2個の規則を合成したものとかんがえられるので，この点について付録で論じる．

ところで，挿入ソートや交換ソートのばあいとはちがって，連結ソートの局所秩序度は金田ら [Kan 93b] の制約充足問題の解法にはしたがっていない．すなわち，0, 1だけでなく，任意の正の実数値をとりうるという特徴がある．もし上記の局所秩序度のかわりに挿入ソートとおなじ局所秩序度をつかったとすると，図10のようにひとつの原子が複数の原子からリンクでさされた状態においても局所秩序度の値はひとしいため反応は停止する．したがって，ただしくソートされない．

同様にリンクを使用する前記の挿入ソートとくらべたとき，連結ソートにはつぎのような特徴がある．

第1に，初期状態において，たとえ逆順のリンクが存在したり，複数のデータからリンクされたがデータが存在したりしても (すなわちデータが単鎖になっていなくても)，ただしくソートされる．あるいは，いったんソートが完了したあとでリンクが逆転されたり，逆順のリンクをもつデータが追加されても同様である．これに対して，挿入ソートにおいては，このようなばあいはただしくソートされない．挿入ソートのばあいには「単鎖である」という制約を保持するために特別のくふうを要していたが，そこではあらかじめこの条件がみたされていないばあいに，それを回復することはできない．

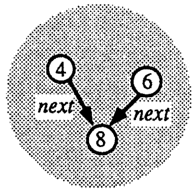


図10　連結ソートの反応規則と挿入ソートの局所秩序度をくみあわせたばあいに生じる不正な結果

第2に，動的に秩序度関数を昇順ソート用から降順ソート用，あるいはその逆にとりかえても，ただしく動作する．

これらの特徴は，連結ソートのキャスタがもつ探索グラフの強連結性 [Kan 93c] という性質からうまれている．これは，インスタンス秩序度に関する条件を無視すれば，適当なデータに関して規則の適用をくりかえすことによって，任意の状態から任意の状態にいけるという性質である．

ところが，連結ソートにはソート法として致命的な問題点がある．それは，同一値が複数個存在するとただしくソートされないということである．上記の局所秩序度は同一値が存在しないことを仮定して定義されていたが，ソートされない理由はそれ以前にある．すなわち，同一値が存在すると，仮に同一値をもつ原子にリンクされた原子の局所秩序度が定義されていたとしても，その値の大小に応じて図11 (a) または (b) のような不正な結果がえられる．

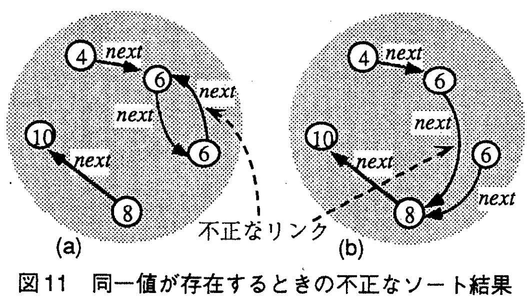


図11　同一値が存在するときの不正なソート結果

すなわち，同一値をもつ原子をむすぶ輪や分岐構造が生成される．この問題は巡回セールスマン問題

において輪が分裂する問題 [Kan 93a] と同根である．この種の問題をなくすには非局所的なデータ参照が必要であり，局所情報だけをつかって計算するという原則をやぶらずに解決する方法はわかっていない．

挿入ソートにおいてはこのような問題はなかったが，それは探索グラフの強連結性がないことと関係している．しかし，この問題をくわしくあつかうことはこの報告の範囲をこえているので，ここではこの関係を指摘するにとどめる．

平均計算時間などの例を図 12 にしめす．これはランダムに生成した整数データをふかさ優先戦略によってソートした例である．この実験においては，初期状態においてリンクは存在しない．ランダム戦略のばあいもほぼ同様の結果がえられる．マッチング回数 (Tests) と計算時間は $O(N^2)$ であり，反応回数はほぼ $O(N \log N)$ であることがわかる．

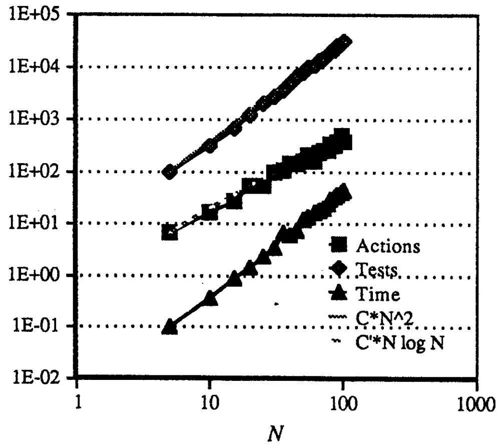


図 12　ランダム・データの連結ソートの計算時間 (ふかさ優先戦略)

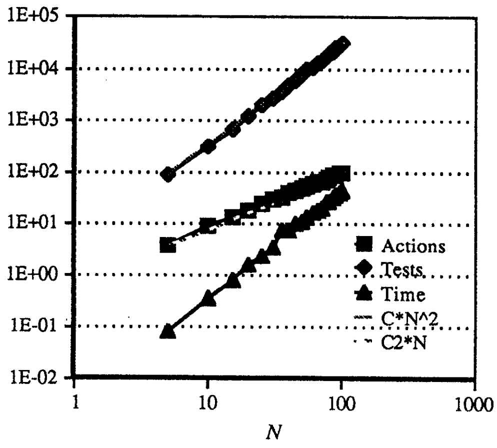


図 13　昇順データの連結ソートの計算時間 (ふかさ優先戦略)

図 13 はふかさ優先戦略をとったときにデータが昇順に選択されるように，あらかじめデータを生成したときの計算時間である．マッチング回数と計算時間に関してはランダム・データのときとかわらないが，反応回数はほぼ $O(N)$ になっている．また，図 14 はふかさ優先戦略をとったときにデータが降順に選択されるようにデータを生成したときの計算時間である．反応回数はほぼ $O(N^2)$ になっている．

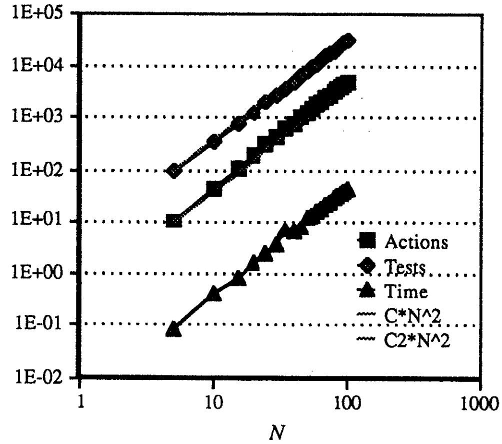


図 14　降順データの連結ソートの計算時間 (ふかさ優先戦略)

6. 選択ソートと $O(N \log N)$ のソート等について

CCM においては局所的な情報だけにもとづく計算を原則としているが，選択ソート (selection sorting [Knu 73]) はこの原則にあわないとかんがえられる．すなわち，選択ソートはすべてのデータのなかから最大値または最小値を選択していくが，この選択においては大域的なデータ参照が必要である．反応規則において不在条件すなわち作業記憶中にある条件をみたす原子がないときになりたつ条件を記述することをゆるせば，選択ソートを記述することができるが，不在条件の使用は上記の原則に反している．最大値・最小値の選択を局所的データ参照からなるステップに分解することができれば，原則をまもって選択ソートを実現することができるだろうが，それを手順的でないやりかたで実現する方法はわかっていない．したがって，選択ソートを CCM によって実現する方法はわかっていない．

また，平均計算時間が $O(N \log N)$ である各種のソートを実現する方法もわからないし，それができるかどうかもわかっていない．適当なデータ構造を定

義すれば，たとえば併合ソート (merge sorting) のプロセスをまねることはできるが，そうしても現在の SOOC のもとでは $O(N^2)$ の時間がかかってしまう．それは，併合ソートやクイック・ソートを実現するためにはデータの値に依存するスケジュリング戦略が必要であり，ランダム戦略やデータの値に依存しない現在の系統的戦略においては，むだな反応はおこらないとしても，むだなマッチングをふせぐことができないからである．

これに対して，bitonic sort のような計算時間が $O(N \log^2 N)$ のソートは実現可能性がよりたかいとかんがえられる．また，$O(N \log N)$ のソートのなかでも，基数ソート (radix sorting) は比較的実現可能性がたかいとかんがえられる．しかし，いずれにしても，これらは今後の課題である．現在は $O(N^2)$ の方法しか実現されておらず，並列性がたかいとしても，CCM によるソートに実用性はない．

7. 結論

CCM は仕様も明確にかきくだせない開放系の問題に適用するために開発したものだが，古典的な問題への適用実験も重要だとかんがえてソート法に適用した．この方法では，唯一のプロダクション規則と唯一の評価関数をあたえるだけで，局所情報の参照だけにもとづいてソートをおこなうことができる．

ただし，局所情報だけによる計算という原則をつらぬこうとすると，同一の値をもつデータが存在するときに線形でないデータ構造が生じる．また現在のスケジュリング戦略によっては併合ソート，クイック・ソートなどの計算時間が $O(N \log N)$ のソートは実現できないという問題がある．

今後は，$O(N \log N)$ あるいは $O(N \log^2 N)$ のソートの検討をつづけるとともに，ソート問題をとくことによってえられた知見を，CCM によって他の問題をとくときにいかしていきたい．

参考文献

[For 81] Forgy, C. L.: *OPS5 User's Manual*, Technical Report CMU-CS-81-135, Carnegie Mellon University, Dept. of Computer Science, 1981.

[Kan 92a] 金田 泰：コンピュータによる自己組織系のモデルをめざして，*第33回プログラミング・シンポジウム報告集*, 1992.

[Kan 92b] 金田 泰：自己組織系としての計算システム — ソフトウェア研究への2つの提案 —，*夏のプログラミング・シンポジウム報告集*, 1992.

[Kan 93a] 金田 泰：プロダクション規則と局所評価関数による最適化，*計測自動制御学会システム工学部会研究会*, 1993.2.

[Kan 93b] 金田 泰，廣川 真男：プロダクション規則と局所評価関数による制約充足問題の解法，*情報処理学会記号処理研究会*, 1993.3.

[Kan 93c] 金田 泰，廣川 真男：プロダクション規則と局所評価関数にもとづく計算モデル CCM による問題解決法の特徴，*SWoPP '93 (情報処理学会人工知能研究会)*, 1993.8.

[Kan 93d] 金田 泰：プロダクション規則と局所評価関数にもとづく計算モデル CCM — その拡張と 0–1 整数計画問題への適用 —，*情報処理学会第47回全国大会*, 1993.10.

[Kan 94] Kanada, Y., and Hirokawa, M.: Stochastic Problem Solving by Local Computation based on Self-organization Paradigm, *27th Hawaii International Conference on System Sciences*, 1994.

[Knu 73] Knuth, D. E., Sorting and Searching, *The Art of Computer Programming*, Vol. 3, Addison-Wesley, 1973.

付録：連結ソートの規則の分解

連結ソートの規則は，図 S1 にしめすような2個の規則に分解することができる．規則 Insert はデータ d1 を始点とするリンクがない状態において適用され，d2 へのリンクを生成する．規則 Redirect はデータ d1 を始点とするリンクがある状態において適用され，リンクがさす先を変更する．すなわち，これらの規則の適用条件は (適用すべきデータがきまれば) 排他的である．図9にしめした規則はこれらの両方の条件のもとで適用され，図 S1 の規則とまったくおなじはたらきをする．規則 Redirect におけるデータ d3 の局所秩序度は，局所秩序度として本文中で定義したものがあたえられれば規則の適用前後で変化しないから，図9の規則と図 S1 の規則とは局所秩序度の計算に関しても等価である．

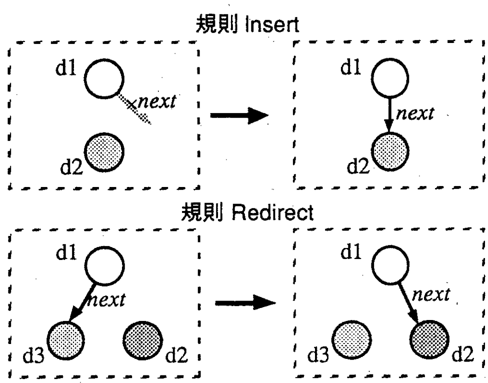


図 S1　2個の規則による連結ソート