自己組織システムとしてのソフトウェア

— 将来のソフトウェア開発への２つの提案 —

日立製作所中央研究所

金田　泰

提案 1

“プログラム” しないようにしよう

■ つまり ...

- ◆「パッケージ・ソフトをつかおう」というようなケチな提案ではない .
- ◆“プログラム” をつかわずにコンピュータ・システムをつかえるようにしよう (研究しよう) .
- ◆ 計算の完全な計画 (決定論的なプログラム) をあらかじめ用意するのはやめよう .

■ なんで？

- ◆ 完全な計画 (バグのないプログラム) をつくることは，しょせん人間には無理．
- ◆ 決定論的な “プログラミング” では “どうでもいい” ことまできちんときめなければならない — むだな努力がつきもの．
- ◆ “プログラミング” はコンピュータ・システムの自己組織能力をころしている (?)
  - 決定論的な世界では自己組織化 (自発的・自律的行為) はおこらない — “プログラミング” をやめることが必要．
  - 非平衡熱力学系，生物系などでは自己組織化がおこる．コンピュータで自己組織化がおこりえない理由があるか ?!

■ どうやって？

◆“ミクロ・モデル”をつくる

部分的な知識だけでソフトウェアが実行できるしかけ(計算モデル)をつくる．

- 部分的な知識だけで＝非決定的な部分をふくんでいても．
- ミクロ・モデルの例はあとでしめす．

◆ “マクロ・モデル”をつくる

システムがのぞみどおりに動作しているかどうかを観測し，制御するしかけ．

―観測と制御のしかけが必要なのは，もはや計算がプログラムによって完全に制御されていはいないから．

ミクロ・モデルとマクロ・モデル

■ ２つのモデルは相補的

- ◆システムのミクロな状態のなかには観測と制御には不要な情報がおおい — 情報圧縮のためにマクロ・モデルが必要．
- ◆ミクロ・モデルからはマクロな状態やふるまいは予測できない (“エントロピー” のようなマクロな状態は記述できない)．
- ◆マクロ・モデルだけではシステムの詳細な動作を決定できない (捨象されている部分があるから)．

■ 自己組織化はミクロ世界とマクロ世界とのからみあいからおこる

- ◆シナジェティクスでもミクロ・レベルとマクロ・レベルとの関係が重視されている (シナジェティクス [Haken] は物理・化学系などの自己組織化の理論)．

提案 2

[自己組織] システムとして “計算” をみよう

■ つまり ...

◆ コンピュータをふくむ “システム” やソフトウェアをシステム理論的にとらえよう .

- ここでいうシステム理論とは , サイバネティクス [Wiener] や一般システム理論 [Bertalanffy] などのこと .

■ なんで？

- ◆ “プログラミング” によらない “計算” においては，計算の観測と制御の理論 (システム理論) が必要である．
- ◆ システムの自己組織化をあたたかくみまもり，その実現をてだすけしよう (そのための理論をつくろう)．

[自己組織] システムとはなにか？*

■ 自然のシステムは４つの特性をもつ [Laszlo]

(“自然のシステム” は自然につくられた社会システムもふくむ)

- ◆ 全体性　　[非還元性]
- ◆ 自己安定性　[“負のフィードバック”]
- ◆ 自己組織性　[“正のフィードバック”]
- ◆ 階層性

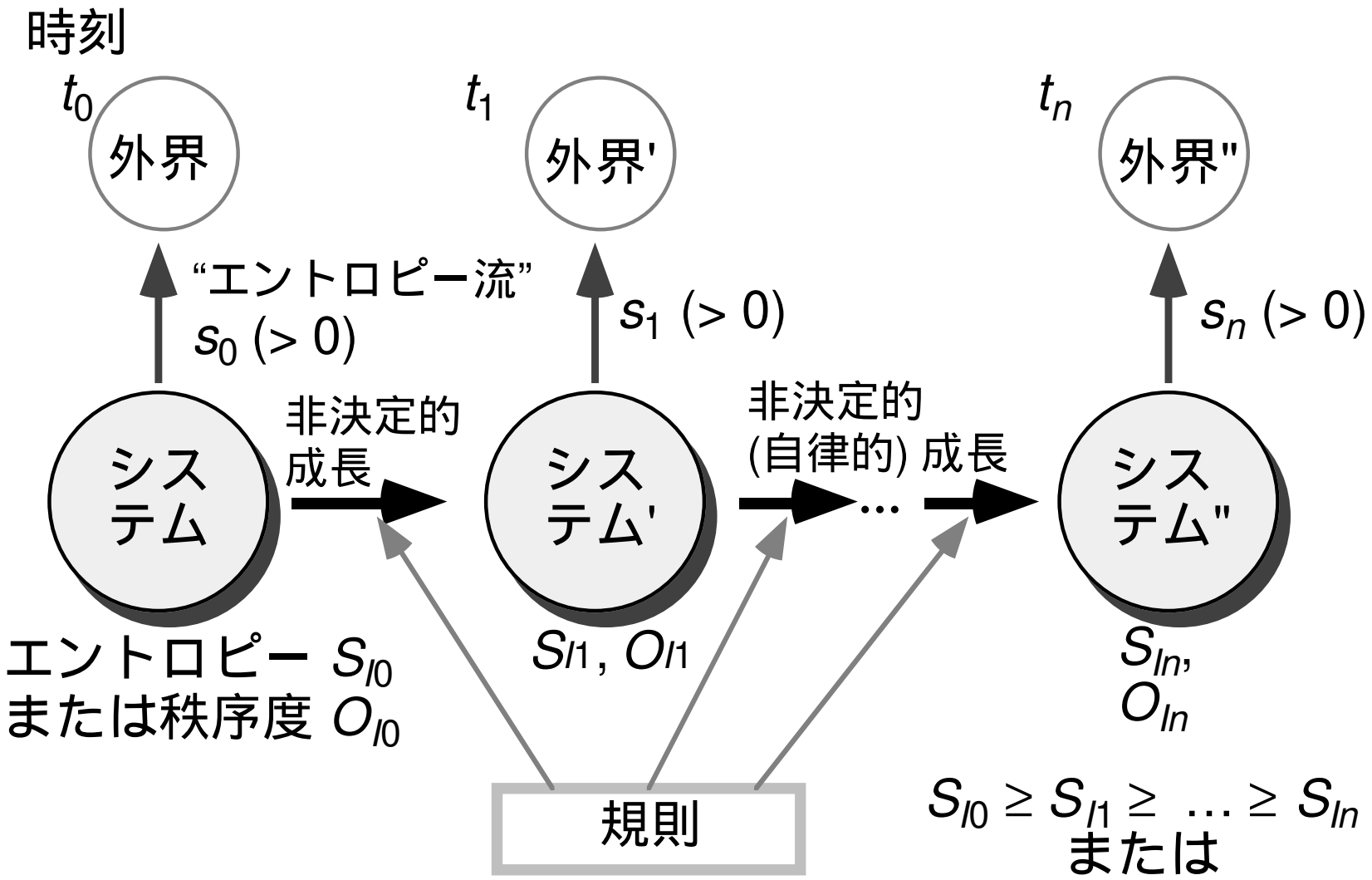
■ 計算システムの非還元性について

- ◆ 従来のソフト開発法は還元論的 (分割統治法的) だが，現実のソフトは非還元的．

■ 計算システムの自己安定性について

- ◆ 些細なバグが致命的なのは自己安定性がないから (?)

自己組織システムの (システム論的) ーモデル



提案１の具体化：計算モデル MCP — 1
MCPの提案

- MCP = 化学的プログラミング・モデル

  (The Model of Chemical Programming)

- MCP は化学反応系とのアナロジにもとづく計算モデルである．

- MCP は不完全な (非決定的な) 計画にもとづく計算のためのミクロ・モデルである．

- MCP は前記の自己組織システムのモデルにもとづいている．

その構成要素

■ オブジェクト (データ)

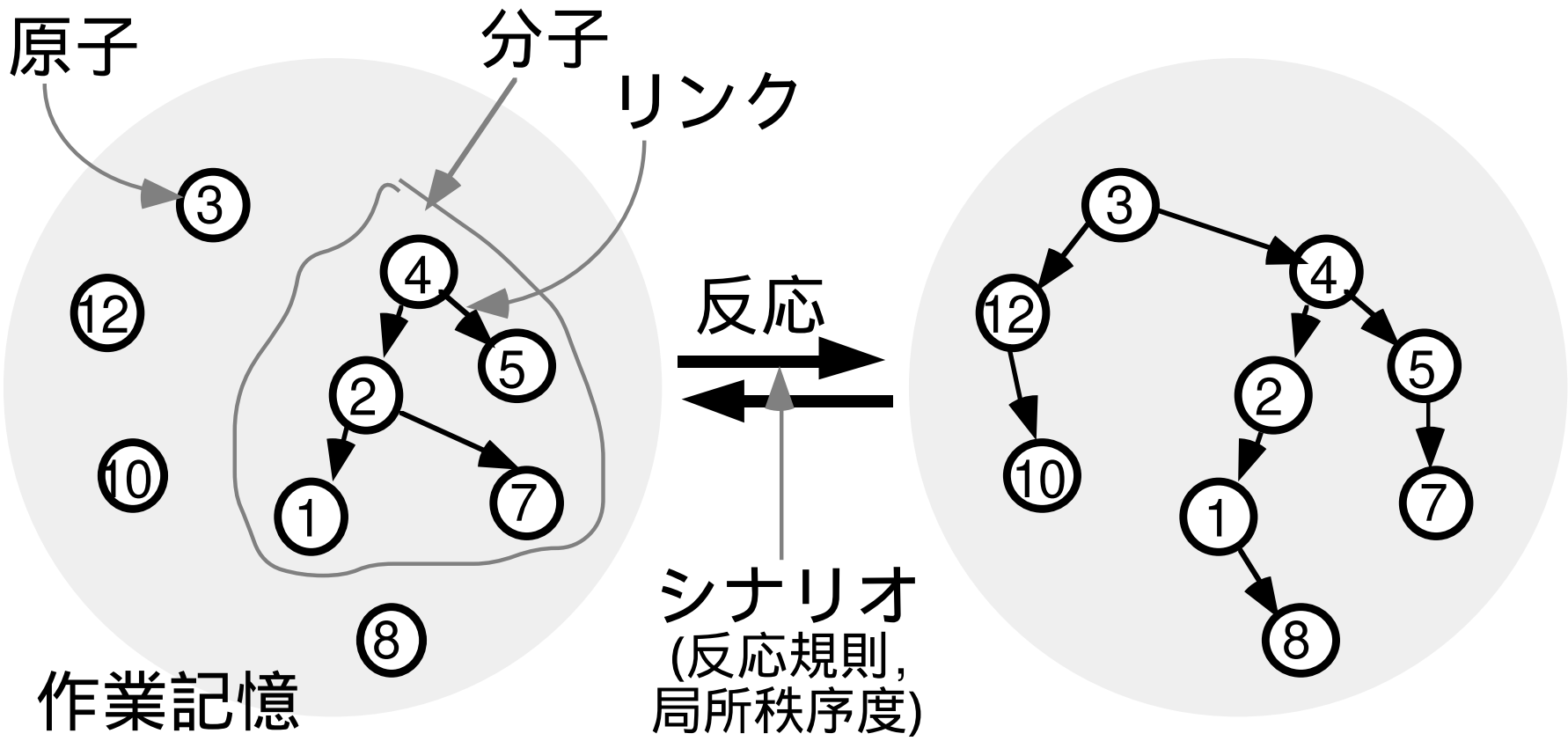
- ◆ 原子： データの単位．原子は内部状態をもつ．
- ◆ 分子： 原子がリンクによって結合されたものである．
- • リンク： 原子を結合するもの．
  向きがあり，ラベルがつく点が化学結合とことなる．

■ シナリオ (プログラムに相当するもの)

- ◆ 反応規則：系の局所的な変化のしかたをきめる規則．
  前向き推論によるプロダクション規則 — 化学反応式に相当．
- ◆ 局所秩序度：局所的な秩序化の程度をあらわす量．

局所秩序度が増加する時だけ反応規則が反復適用される．
局所的な“計画”による大域的な“組織化”をめざす．

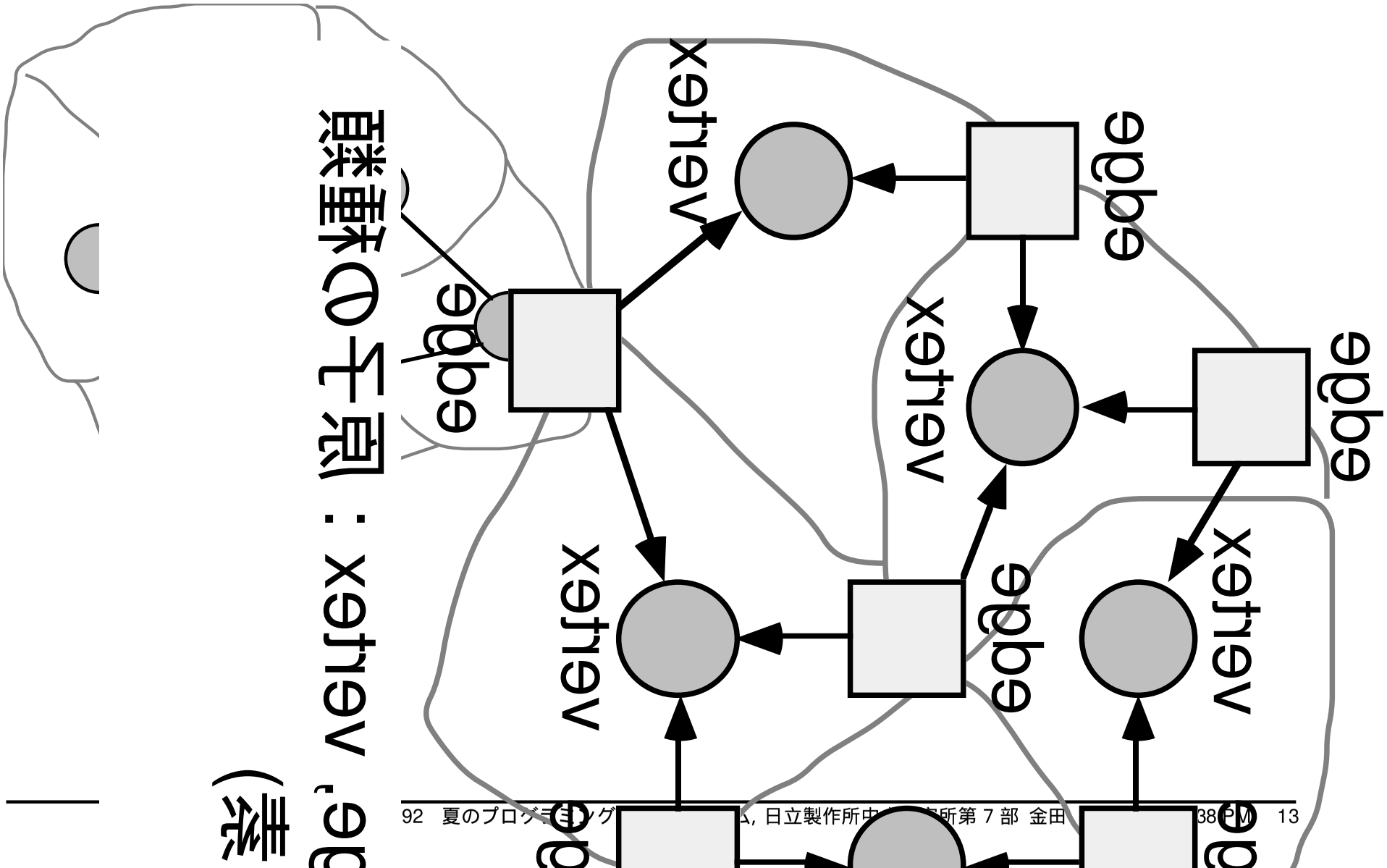
MCP の構成要素の図説



◆ 規則の適用順序は非決定的 —どの規則とどのデータから反応させるかはシステムが自律的にきめる．

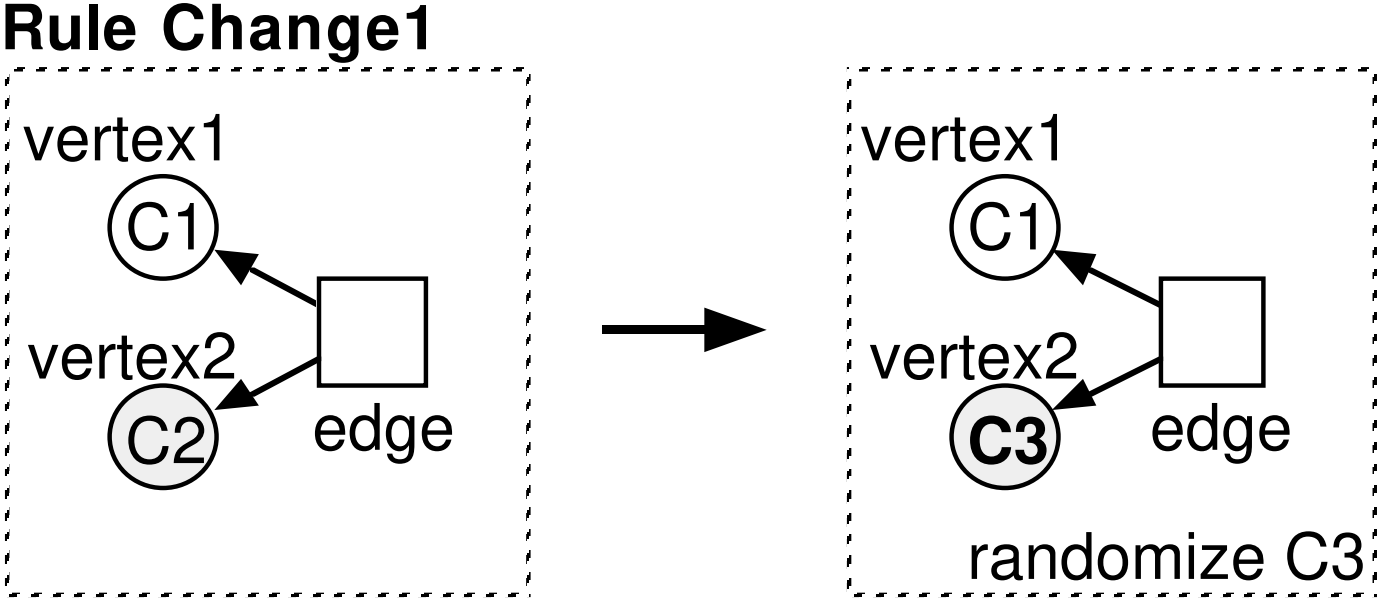
例題：グラフ彩色問題

問題　　データ表現



グラフ彩色問題のシナリオ — 1
反応規則の定義

■ 隣接２頂点の一方をランダムにぬりかえる規則



あらかじめグラフを表現するデータがあたえられていれば，
規則は１個だけでよいとかんがえられる．

局所秩序度の定義

■ 元素 edge の局所秩序度定義

$$o_{edge}(x) = \begin{cases} 1 & \textbf{if } x.vertex1.color \neq x.vertex2.color \\ 0 & \textbf{otherwise} \end{cases}$$

すなわち，辺の両端の頂点が同色ならば 0，そうでなければ 1．

■ 元素 vertex に対する局所秩序度は定義しない

(局所秩序度は一定とする)．

■ vertex2 が規則適用前の vertex1, vertex2 とことなる色にぬりかえられるときだけ，規則が適用される (規則にはこの条件を記述する必要はない)．

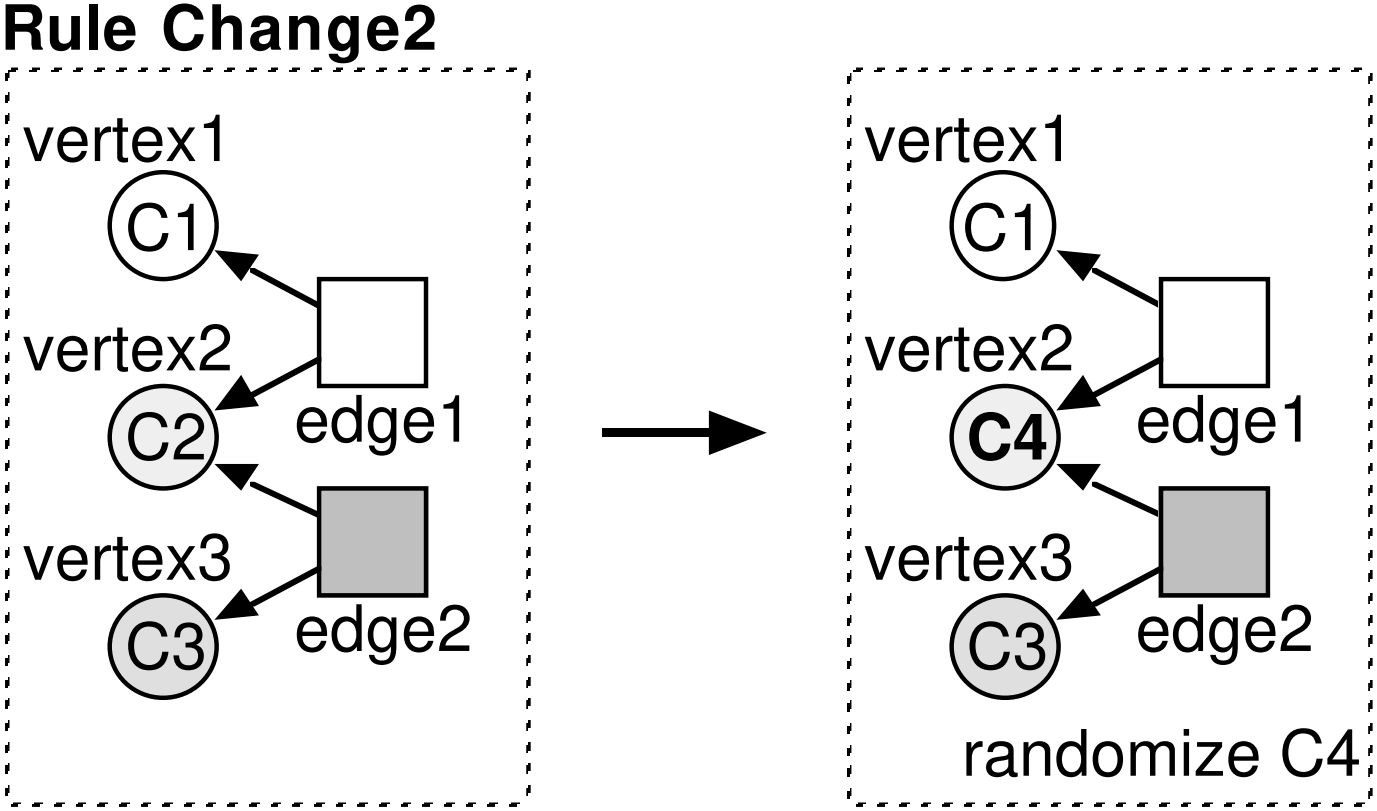
グラフ彩色問題の実行例 — 1

- MCP にもとづく言語 SOOC とその処理系を作成した (SOOC = Self-Organization-Oriented Computing) .
- 前記のシナリオをそのまま実行させると，非常に非効率 .
  - たとえば，ある 17 頂点，46 辺のグラフは 1000 回以下の規則適用では解に到達しなかった .
- 規則をより非局所的なものにとりかえると，そのグラフで効率よく解をもとめることができた .

―2 *

■ より非局所的な規則の例

Rule Change2



規則 Change2 の動作は基本的に Change1 とおなじだが，よりひろい範囲のデータを参照して動作する点がことなる．

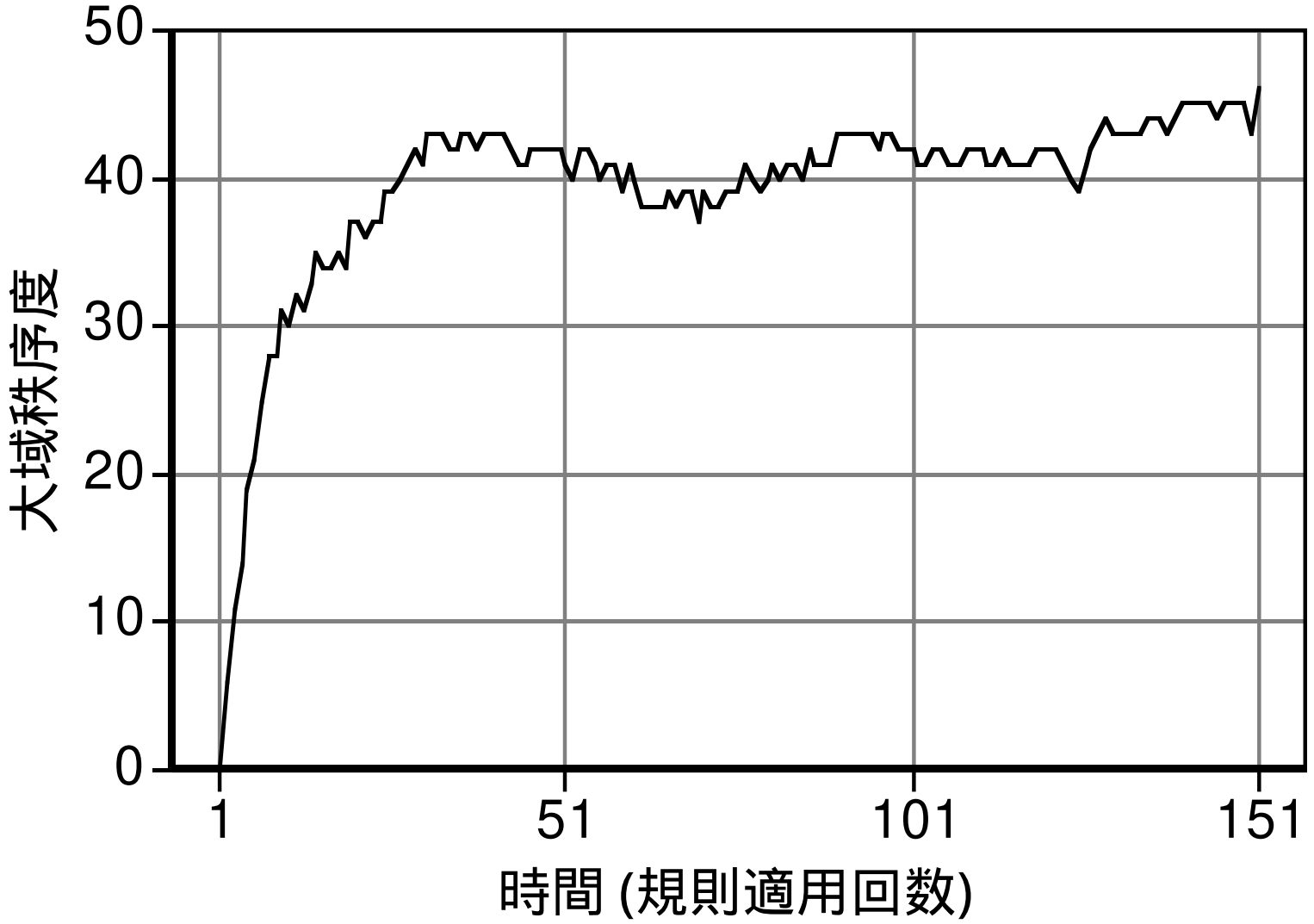
提案２の具体化：計算を確率過程としてみる — 1
大域秩序度の定義

■ MCP における局所秩序度の作業記憶全体にわたる和を大域秩序度という．

◆ グラフ彩色問題のばあいは，すべての辺の局所秩序度の和が大域秩序度．

◆ 辺の総数を N とすると，大域秩序度 O の値の範囲は $0 \leq O \leq N$.

グラフ彩色における大域秩序度の実測例



マルコフ連鎖にもとづくマクロ・モデル

- 大域秩序度の時系列をマルコフ連鎖とみるマクロ・モデルを構成した．
- 大域秩序度の変化をうまく説明できるモデルがえられた．
  - このモデルからの予測が実測とよく一致した．
    - 計算時間の分布のかたちが一致．
    - 計算開始時から，強非定常状態 → 準定常状態 → 定常状態 (解に到達した状態) のように遷移する．

グラフ彩色問題以外の実験 *

■ N クウィーン問題

- ◆ ある条件のもとで $N=50$ まで実験し，計算時間は $O(N^{4.6})$ であることがわかった．

■ 巡回セールスマン問題 (TSP)

- ◆ 都市数 $N=500$ まで実験し，計算時間は $O(N^3)$ 程度であることがわかった．

■ 線形計画問題・整数計画問題

■ ソート

...

まとめ

■ ソフトウェア開発に関連して２つの提案をした．

- ◆ 提案 1: “プログラミング” しないようにしよう．
- ◆ 提案 2: [自己組織] システムとして “計算” をみよう．

■ 提案１を具体化するために

- ◆ MCP という計算モデルをしめした．
- ◆ 例題としてグラフ彩色問題をとりあげて，MCP にもとづいて問題がとけることを実証した．

■ 提案２を具体化するために

- ◆ MCP にもとづく計算を確率過程 (とくにマルコフ連鎖) とみる見方をしめした．