線型論理による計算の並列度向上法

4X−4

中沢 健 新潟大学大学院自然科学研究科
澤村 一 新潟大学工学部情報工学科

1 はじめに

並列計算モデルにおいては待ち状態を含む計算効率の悪い計算過程，待ち状態を含まない計算効率の良い計算過程がある．前者を並列度の低い計算，後者を並列度の高い計算と呼ぶ．

本論文ではペトリネット [1] で与えられた並列計算モデルの計算過程を記号論理学の１つである線型論理を用いてより並列度の高い計算へと変換する方法について述べ，さらに変換を計算機上で実現するためのデータ構造及びそれを用いた変換方法について述べる．

2 計算の並列度

並列計算モデルの例として図 1のペトリネットを考えることにする．ここでは，○が状態，□が計算処理，トークンが計算資源を表す．

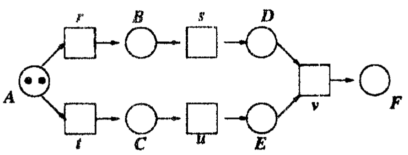

図 1: 並列計算モデル

図 1のモデルではトークンが Fに移動した時点で計算終了となる．その際には $((r||t);(s||u);v)$ または $((r;s)||(t;u));v$のように複数の計算過程が存在する．前者の計算過程はｒとｔ，ｓとｕを並列に計算するため待ち状態が生じ並列度は低い．逆に後者の計算過程は待ち状態がなく並列度は高いと言える．

3 計算モデルと線型論理の関連付け

線型論理とは論理概念に資源の概念の入った論理学である．ここでは線型論理の表記方法でモデルおよび，その計算過程の表現を考える．

$A \vdash B$は「ＡからＢを得ることができる」を意味する．同様に考えるとモデルは５本の式から成る式 (1) で表現できる．

$$\begin{matrix} A \vdash B & B \vdash D & \\ & & D \otimes E \vdash F \\ A \vdash C & C \vdash E & \end{matrix} \qquad (1)$$

これらを証明の始式として，以下に示す推論規則 $\otimes R, \otimes L$, カット を用い結論 $A, A \vdash F$を導く計算の証明木を作成すると並列度の低い計算，高い計算はそれぞれ図 2で示される．

- ⊗R
$$\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}$$
- ⊗L
$$\frac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C}$$
- カット
$$\frac{\Gamma \vdash A \quad \Delta, A \vdash B}{\Gamma, \Delta \vdash B}$$

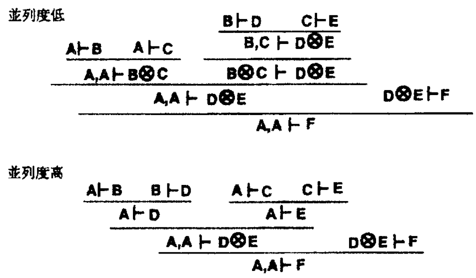

図 2: 証明木

ここで推論規則カットに注目してみる．推論規則カットには本質的カットと本質的でないカットの２種類がある．カットされる式がネット式であるカットが本質的カット，そうでないカットが本質的でないカットである．ここでネット式とは $X \vdash Y$の Xに現れる式である．図 1のモデルでは $A, B, C, D \otimes E$がネット式となる．図 2より本質的でないカットを含む証明木が並列度の低い計算，逆に本質的でないカットを含まない証明木が並列度の高い計算である．

---

[0]Raising degree of concurrency by linear logic , Ken Nakazawa, Graduate School of Sience and Technology, Niigata University , Hajime Sawamra, Dept. of Information Engineering, Faculty of Engineering, Niigata University

4 証明変換

証明変換とは証明木内から本質的でないカットを除去することである．除去には線型論理の変換規則を用いる．変換規則は公理，置換，ロジックと大きく分けて3種類，合計9種類与えられており[3]，用いる規則は本質的でないカットの上式に用いられている推論規則によって定まる．本質的でないカットの発見，変換規則の適用の操作を繰り返し行ない証明木内に本質的でない カット がなくなったら変換の終了である．

5 計算機上での実現

これまで述べてきた変換操作を図3で示す手順で計算機上で実現する．

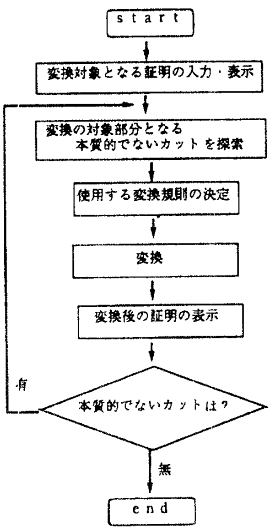

図 3: 証明変換流れ図

まず証明木の計算機上での表現のために図4で示すような，推論規則の種類，上式左側，上式右側，下式をメンバとする構造体1，左辺，右辺，左辺と右辺を結ぶ演算子をメンバとする構造体2，演算子または項を表す構造体3と3つの構造体を定義する．

次に証明変換の方法を考える．証明の入力方法は，まず構造体1を与えて用いられた推論規則，下式として構造体2と構造体3を用いて証明木の結論となる論理式を入力する．続いて上式左側に新たに構造体1をつなぎ同じ手順で下式と推論規則を入力する．この操作を続け，上式左側に構造体をつなぐ必要がなくなったら上式右側にも同様にして構造体1をつなぎ用いられた推論規則と論理式を入力する．

こうして作られた証明木の木構造から全探索を行ない変換の対象部分を求める．変換規則は変換対象部分の上式左側，上式右側に用いられている推論規則を調べて定める．変換の操作については変換規則と同様になるような木構造の組み変えで実現される．これらの操作を証明木内から本質的でないカットがなくなるまで繰り返す．

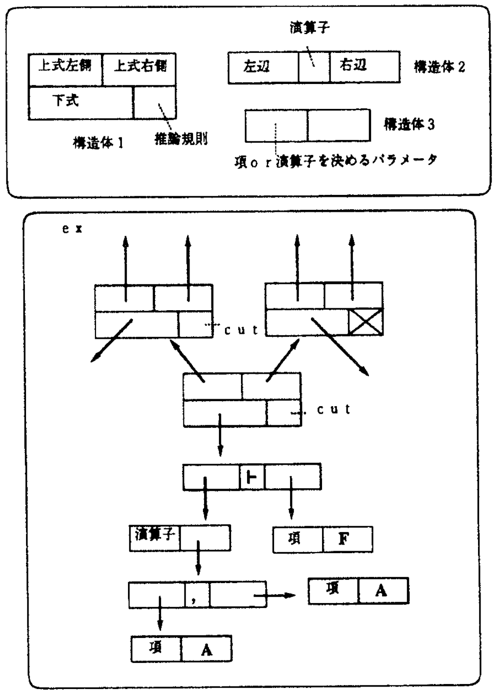

図 4: データ構造

6 まとめ

証明木を計算機上で表現するために3種類のデータ構造を定義した．このデータ構造によって証明変換は計算機上で実行でき，手計算で行なうのは困難である証明変換も実現可能となる．今後は，ペトリネットに時間の概念が入った場合の変換方法を考えていく．

参考文献

[1] W. Reisig:"Petri Nets", Springer-Verlag, pp.3-16,1985.

[2] 竹内外史:"線型論理入門", 日本評論社, 1995

[3] C. Gunter and V. Gehlot: "Nets As Tensor Theories", Application and Theory of Petri Nets, pp. 174-191