

4V-8

ネット接続性検証における分割処理手法の実現と評価

須藤智裕¹⁾、山田純世¹⁾、松尾ゆり²⁾、飯塚一俊²⁾、肥後野恵史²⁾

東芝CAEシステムズ株式会社¹⁾、(株)東芝 総合情報システム部²⁾

1.はじめに

一般的に多用されているパストレース法によるネットの接続性検証では、それに要するメモリ量及び処理時間が、そのネットに所属する全要素(ワイア、ビア、ピン)数の2乗に比例する。従って、長い(要素数が多い)ネットでは、メモリ量、処理時間も非常に大きくなり実用的ではない。そこで、この問題を解決するため分割処理手法を取り入れたアルゴリズムを考案、導入する。本アルゴリズムは、検査領域を分割し、その領域毎の検証を行い最後にそれらの結果を併合して、ネット全体の接続性を検証するという手法を用いることで、必要なメモリ量の削減および処理時間の短縮を可能としている。

2.アルゴリズムの概要

検査領域を適当に分割し、その分割領域毎に検証を行い、全ての分割領域での検証結果を基にネット全体の接続性を調べる。以下にその手順を示す。

- 1) 検査領域を分割する。
- 2) 対象のネットに所属する要素の中で、境界線と交差する(2つ以上の分割領域にまたがる)ものを全てノードに登録する。(ここで登録されるノードを境界ノードと呼ぶ)
- 3) 境界ノード同志の接続関係を記録した接続マップを作成する。全ての分割領域について以下の4)～7)を繰り返す。
- 4) その分割領域に完全に含まれる(境界線と交差しない)ノードを全てノードに登録する。(ここで登録されるノードを分割ノードと呼ぶ)
- 5) 境界ノードと分割ノード及び分割ノード同志の接続マップを作成する。

- 6) 5)で作成された接続マップを基に、その分割領域と交差する全ての境界ノードを開始のノードとしてパストレース法を行う。パストレースの途中で開始のノードと異なる境界ノードが現れたら、逐次接続マップに記録(更新)する。
- 7) 全ての分割ノードがトレース済みかどうかを調べる。全ての分割ノードがトレース済みならば可とし、トレースされていない分割ノードが1つでもあれば不可とする。(不可ならばそのネットはOPENであるので処理をやめる)
- 8) 6)で更新された境界ノードの接続マップを基に、任意の境界ノードを開始のノードとしてパストレース法を行う。
- 9) 全ての境界ノードがトレース済みかどうかを調べる。全ての境界ノードがトレース済みならば可とし、トレースされていない境界ノードが1つでもあればそのネットはOPENと判定される。

3.効果

本アルゴリズムの効果は、境界線の引き方に依存している。しかし、実際の検証で、ネット毎に境界線を作り替えていたのでは、それに要する処理時間の増大という新たな問題を生み出してしまふ。そこで、その回避策として、全てのネットに対応する単純な(例えば水平または垂直方向のみの)境界線を定めておくなどの前処理が必要であり、また、有効な方法でもある。

Implementation and evaluation of net connectivity verification algorithm with divided method

Tomohiro SUDOH, Sumiyo YAMADA
Yuri MATSUO, Kazutoshi IIZUKA, Shigefumi HIGONO

TOSHIBA CAE Systems, Inc.
TOSHIBA Corp.

4. 評価

今回は、境界線を水平方向のみとし、まずその本数を1～50の間で変化させて各分割領域毎の要素数を調べ、最適な境界線数を求めた。その結果、データの大きさ(ネットの本数)にかかわらず、境界線の数が5～10の時、境界要素数と最大の分割要素数の和が全要素数の約2分の1程度となり最小になることがわかった。そこで次に、境界線の数を9(分割領域が10領域)にしてメモリ量及び処理時間を調べた。その結果を以下に述べる。

4.1 メモリ量

ネットの全要素数と接続性検証に必要なメモリ量との関係を図1に示す。図1から、要素数の多いネットほど本アルゴリズムの効果が大きい、即ち、検証に必要なメモリ量の差が大きいのことがわかる。この理由は、使用メモリ量が、ノード数に比例するものとノード数の2乗に比例するものととの和である為である。

4.2 処理時間

処理時間の測定に当たっては、ネット1本毎の検証時間が非常に短く、意味のある数値が得られなかったため、実際にいくつかのテストデータを検証して、旧アルゴリズムのそれと比較した(表1)。ここでわかることは、データが大きい程、アルゴリズムの効果も大きいということである。この理由としては、一般に、大きなデータほどネット1本当りの平均の要素数が多いということが挙げられる。

表1 処理時間に関する新/旧アルゴリズムの比較

ネット数	旧アルゴリズム		新アルゴリズム	
	T.A.T.	CPU	T.A.T.	CPU
150	2.0	0.8	1.0	0.5
784	36.0	12.4	13.0	4.9
5742	540.0	174.7	150.0	50.0

5. まとめ

ここに記述したアルゴリズムは、データの大型化に伴うメモリ量の増大と処理時間の増加という問題に対する1つの回答として導出された。そして、この2つの問題を同時に解決し十分な効果が上がることも確認された。今後は、このアルゴリズムの採用と併せて、更に効果的な検証方法を検討したい。

参考文献

- T.Ohtsuki(editor) : LAYOUT DESIGN AND VERIFICATION, Elsevier Science Publishes B.V. (North-Holland), 1986

(k -bytes)

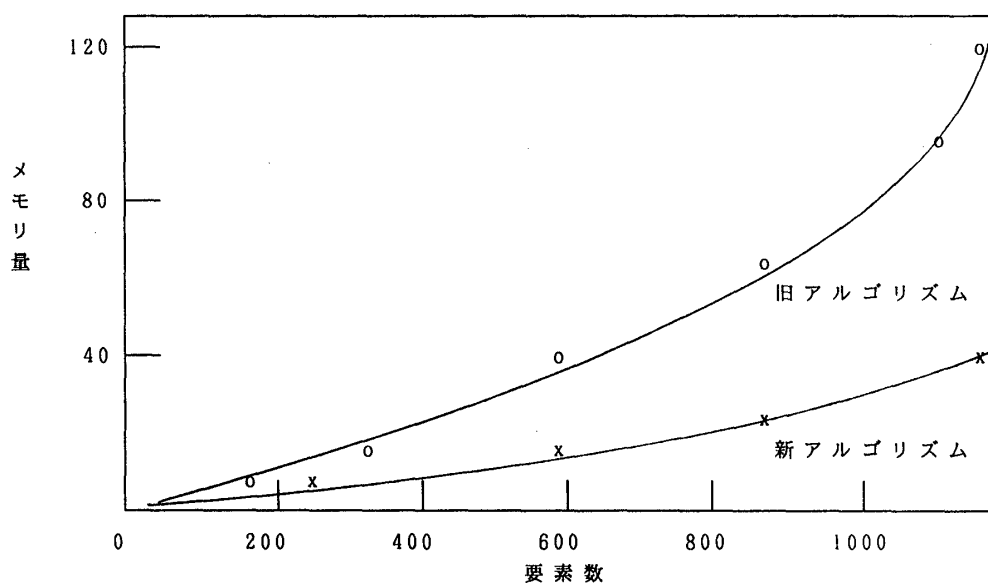


図1 ネットの全要素数とメモリ量の関係