データ依存回路による部分グラフ同型判定

指導教官：市川 周一　　学籍番号 003317　山本昌治

1　はじめに

部分グラフ同型判定は化学物質の構造活性推測やシーン解析など幅広い応用を持つが, 一般に NP 完全であり計算は困難である. 広く用いられている Ullmann のアルゴリズムは論理回路で実現し高速に判定できる [1] が, 回路規模は過大で実装が困難である [2].

本研究では, データ依存の論理回路を実装することにより Ullmann の回路の規模を縮小する手法について検討した. 評価には Lucent OR2C FPGA を使用し, 1. 回路規模, 2. 演算性能, 3. 回路生成時間について評価した.

2　Ullmann の回路

2 つのグラフ G_α と G_β が与えられたとき, G_α が G_β の部分グラフと同型であるか判定する問題を部分グラフ同型判定問題という. 例えば図 1 において, G_β は G_α と同型な部分グラフを持つが G_γ は持たない.

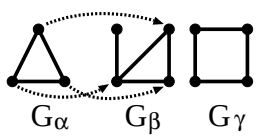

図 1: 同型判定の例

Ullmann は図 2 の要素回路を $p_\alpha \times p_\beta$ 個並べた判定回路を提案した (p_α, p_β はそれぞれ G_α, G_β の頂点数)[1]. 要素回路の入力 a_{iy} $(1 \le i, y \le p_\alpha)$, b_{xj} $(1 \le j, x \le p_\beta)$ はそれぞれ G_α, G_β の隣接行列 A, B の要素をあらわし, m_{ij} $(1 \le i \le p_\alpha,\ 1 \le j \le p_\beta)$ は写像 M の要素をあらわす. A, B, M はメモリとして実現される. Ullmann の回路の論理規模は $O({p_\alpha}{p_\beta}^2)$ で増加する [2].

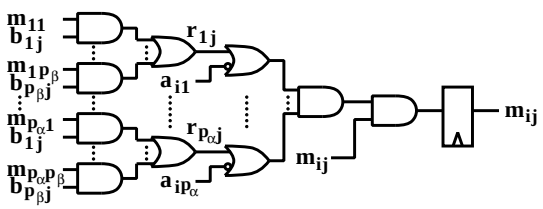

図 2: Ullmann の要素回路

3　データ依存回路

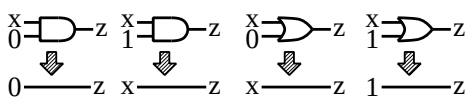

図 3: 簡単化の例

論理回路の入力が定まると, 図 3 に示すように論理を簡単化してゲート数を減らすことができる. 例えば入力 G_α に応じて隣接行列 A が決まれば, 図 2 の a_{iy} が定数になり論理が簡単化される (メモリ A も不要になる). 本研究では, このような回路をデータ依存回路と呼ぶことにする.

データ依存回路は入力グラフ毎に回路を生成するので, 1. 入力グラフに対応する回路記述生成, 2. 論理合成, 3. テクノロジ・マッピング, 4. 配置配線, 5. FPGA 上での判定という段階が必要になる. なお, 論理の簡単化は, 回路記述生成と論理合成の段階で自動的に行われる.

4　設計

INDEP　2 節で説明した Ullmann の回路.

ALPHA　図 2 の a_{iy} を入力 G_α に応じた定数に置き換えて生成した回路. メモリ A を削除できる.

BETA　図 2 の b_{xj} を入力 G_β に応じた定数に置き換えて生成した回路. メモリ B を削除できる.

BOTH1　図 2 の a_{iy} と b_{xj} の両方を入力 G_α と G_β に応じた定数に置き換えて生成した回路. メモリ A と B を削除できる.

BOTH2　図 2 の m_{ij} の初期値により BOTH1 を改善した回路. アルゴリズムの性質として, 一度 0 になった m_{ij} は再び 1 にはならない [1]. この性質を利用して初期値が 0 の m_{ij} を 0 に置き換えた回路で, メモリ M が減少する.

5　評価

本手法では疎なグラフに対して大きな効果があると思われるので, ランダムに生成した木を入力グラフとして用い, 前述の 5 つの回路を評価した. 論理合成には Synopsys FPGA Compiler II を Duron 800MHz PC 上で, テクノロジ・マッピングには Lucent ORCA Foundry 9.4a を Pentium II 450MHz PC 上で実行して, 回路規模 (PFU 数) と動作周波数を見積もった. 次に C 言語で実装したシミュレータで同型判定に必要なクロック数を求め, クロック数と動作周波数から回路での判定時間を見積もった. 性能の比較対象として Ullmann のアルゴリズムを C 言語にて実装し, Pentium III 600MHz PC 上での判定時間を測定した. 回路の生成時間の評価には, 論理合成とテクノロジ・マッピングの時間を用いた (本研究では配置配線は行っていない). 回路規模と判定時間は入力グラフに依存するので 50 組の平均より評価した.

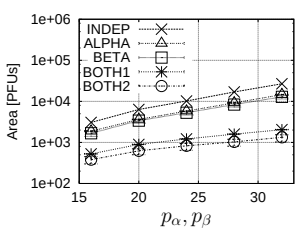

図 4: 回路規模

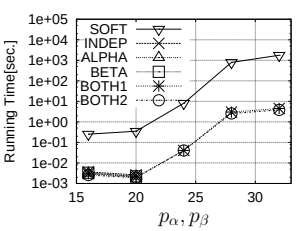

図 5: 判定時間

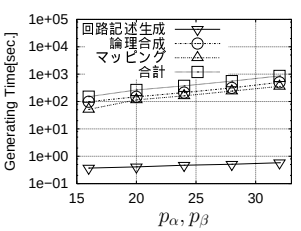

図 6: BOTH2 の生成時間

p_α, $p_\beta \ge 24$ では, マッピング・ツールの異常終了のため INDEP, ALPHA, BETA の判定時間を見積もることができなかった.

$p_\alpha = p_\beta = 32$ のとき, BOTH2 の回路規模は INDEP の 1/20(図 4), BOTH2 の性能はソフトウェアの 460 倍 (図 5) である. BOTH2 の生成時間は約 15 分, 判定時間は 10 秒以下で, 合計でもソフトウェアの判定時間の半分程度である (図 5, 図 6).

6　おわりに

BOTH2 では回路規模が 1/20 になり, 回路生成のオーバーヘッドを考慮しても判定時間を短縮できる可能性がある. 本研究では汎用の回路合成ツールを使用したが, データ依存回路の生成時間は, 問題に特化したツールを作成し利用すれば短縮可能であろう.

参考文献

[1] J.R.Ullmann: “An Algorithm for Subgraph Isomorphism,” J.ACM, Vol.23, No.1, pp.31–42(1976).

[2] S.Ichikawa et al.: “Evaluation of Accelerator Designs for Subgraph Isomorphism Problem,” FPL 2000, LNCS 1896, Springer, pp.729–738(2000).