An implementation method for the Subgraph Isomorphism Judgement Algorithm using FPGAs

Graduate Advisor: Shuichi Ichikawa Registration Number: 931709 Name: Kouji Konishi
Keywords: Subgraph Isomorphism, NP Complete, FPGA, OR2C, OPERL board

1. Introduction

The subgraph isomorphism problem has wide application[1]; for example, the retrieval of chemical bonds from chemical structural formulae, the scene analysis for pattern recognition, etc. However, the subgraph isomorphism problem is NP complete[2] in general, hence no efficient algorithm exists.

This thesis presents a method to solve the subgraph isomorphism problem by using FPGA (Field Programmable Gate Array) for acceleration. With FPGAs, custom architectures can be implemented at a low cost[3]. The OPERL board[5], which is a reconfigurable logic board for PCI bus, is used in this study. Currently the implementation haven't completed, so the proposed algorithm and the performance estimation are described.

2. The Subgraph Isomorphism Problem

A graph is composed of the vertex set V of $p(\geq 1)$ vertexes, and the edge set E of $q(\geq 0)$ edges, and denoted as $G = (V, E)$. Using $v_i, v_j \in V$, an edge $e_k \in E$ can be represented as $\{v_i, v_j\}(i \neq j)$. Non-directed graph is assumed throughout this thesis.

Consider two graphs; $G_\alpha = (V_\alpha, E_\alpha)$ and $G_\beta = (V_\beta, E_\beta)(p_\alpha \leq p_\beta$, where p_α is $|V_\alpha|$, p_β is $|V_\beta|)$. When G_β contains a subgraph isomorphism to a G_α, there is a one-to-one function $f : V_\alpha \rightarrow V(\subseteq V_\beta)$, where for all $\{v_{\alpha i}, v_{\alpha j}\} \in E_\alpha$, there is $\{f(v_{\alpha i}), f(v_{\alpha j})\} \in E(\subseteq E_\beta)$. The problem to find out f, which satisfies the condition, is called as "the subgraph isomorphism problem." Also, the algorithm to solve the subgraph isomorphism problem is called as "the subgraph isomorphism judgement algorithm."

3. The Search Space Reduction

This section explains how to reduce search space for two algorithms; the proposed algorithm and refinement procedure[1]. In both algorithms, the depth first search is used for touring the search tree. The ith depth $(1 \leq i \leq p_\alpha)$ of the tree corresponds to $f(v_{\alpha i})$.

In the matrix $M = [m_{ij}]$ $(1 \leq i \leq p_\alpha, 1 \leq j \leq p_\beta)$, $m_{ij} = 1$ means that $f : v_{\alpha i} \rightarrow v_{\beta j}$ is possible. The initial value of M is decided in this way; if $deg(v_{\alpha i}) \leq deg(v_{\beta j})$ then $m_{ij} = 1$, in another case $m_{ij} = 0$. $deg(v)$ is the degree of vertex v. $A = [a_{ij}]$, $B = [b_{ij}]$ are the adjacency matrix of G_α, G_β, respectively.

In the proposed algorithm, the search space reduction is executed as follows. At the dth depth, decide the corresponcence of $v_{\alpha d}$ to $v_{\beta j}$ where $m_{dj} = 1$. If the correspondence doesn't satisfy $\forall i(i < d)(a_{id} = 1 \rightarrow b_{f(i)j} = 1)$, the case won't be a subgraph isomorphism, therefore no need to tour under that node. On the other hand, the refinement procedure investigates the condition: $\forall x(1 \leq x \leq p_\alpha)((a_{ix} = 1) \rightarrow (\exists y(1 \leq y \leq p_\beta)(m_{xy} \cdot b_{yj} = 1))$ for all i, j. If the case doesn't satisfy, make $m_{ij} = 0$ and repeat this procedure until M stays unmodified.

4. Required Amount of Resources for Proposed Algorithm

OPERL board has Lucent ORCA OR2C series FPGAs[4] on it. OR2C contains a 2-dimensional array of PFU (Programmable Function Unit), and 900 PFUs are available in OR2C40A. The total number of PFU required for the proposed algorithm is shown in Figure 1. p_α, p_β and ed_α are parameters. ed_α is the edge density of G_α, which is the proportion of q_α ($q_\alpha = |E_\alpha|$) to $p_\alpha(p_\alpha - 1)/2$ (the number of edges in the complete graph with p_α vertexes). Figure 1 shows the estimated number of PFUs at $ed_\alpha = 0.50$ in the proposed algorithm. The number depends on $\max(O(p_\alpha^2 \cdot \log p_\alpha), O(p_\beta^2))$. Small graphs such as $p_\alpha + p_\beta \leq 110$ are expected to be implemented by an OR2C40A FPGA. When $p_\alpha + p_\beta$ is small enough, OR2C40A can contain two or more set of required hardware to split the search space to search isomorphism in parallel.

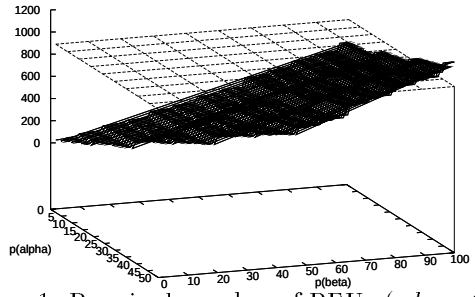

Figure 1: Required number of PFUs ($ed_\alpha = 0.50$)

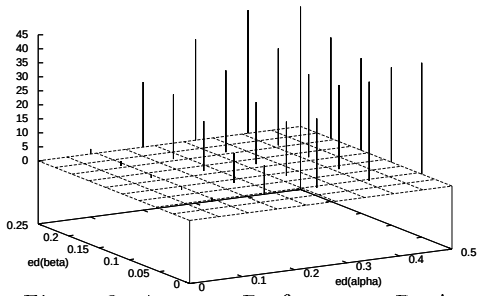

Figure 2: Average Performance Ratio

The resource required for refinement procedure is at least $O(p_\alpha^2 \cdot p_\beta^2)$, which is unpractical for hardware implementation. Contrary, the search space reduction in the proposed algorithm is simple enough to make hardware implementation practical.

5. Performance Comparison

The OR2C on OPERL board works at 33 MHz. The execution time of the proposed algorithm is estimated by summing up the required clocks for each step in the proposed algorithm. The execution time of refinement procedure implemented by C language is also measured on the computer, which has Pentium-II 333 MHz CPU, 512 KB 2nd cache, 128 MB main memory with FreeBSD-2.2.6-RELEASE.

In both cases, the measurement is done in this way; at all case $1 \leq p_\alpha \leq 10, 1 \leq p_\beta \leq 15 (p_\alpha \leq p_\beta)$, G_α and G_β are given 50 times at random. In case $p_\alpha + p_\beta$ is small, the parallel hardware implementation is adopted for FPGA. Figure 2 shows the average performance ratio, where the measurement is done in $0.1 \leq ed_\alpha \leq 0.5, 0.05 \leq ed_\beta \leq 0.25$. The greater ed_α and ed_β, the better the performance of the proposed algorithm.

6. Conclusion

Hardware-software co-design of subgraph isomorphism problem is considered feasible with the state-of-the-art FPGA.

From now on, the proposed algorithm is planned to be implemented onto OPERL board for performance evaluation. The extension of the proposed algorithm is also planned by utilizing the dynamic reconfiguration of OR2C FPGA.

References

[1] J. R. Ullmann, "An Algorithm for Subgraph Isomorphism," Journal of the Association for Computing Machinery, Vol. 23, No. 1, pp. 31-42, 1976.

[2] Michael R. Garey, David S. Johnson, "Computers and intractability: a guide to the theory of NP-completeness, " W. H. Freeman, San Francisco, 1979.

[3] Duncan A. Buell et al., "Splash 2: FPGAs in a Custom Computing Machine," IEEE Computer Society Press, Los Alamitos, 1996.

[4] Lucent Technologies Inc., "ORCA^{TM} OR2CxxA and OR2TxxA series field programmable gate array," Oct. 1997.

[5] Shuich Ichikawa, Toshio Shimada, "The trial and the evaluation for the reconfigurable board added to PCI bus," IEICE tech. reports CPSY96-97, pp. 159-166, 1996.