

## 発表概要

## 部分冗長除去に基づく実践的な大域命令スケジューリング

及川 亮太郎<sup>1,a)</sup> 滝本 宗宏<sup>1</sup>

2012年3月16日発表

命令レベル並列性を高める有効な手法の1つとして、命令スケジューリングがある。特に、投機的実行を許す命令スケジューリングは、さらに並列度を高められる点で効果的であることが知られている。投機的実行は、元のプログラムに存在しなかった冗長性を導入する可能性があるため、1つの命令をスケジュールするたびに共通部分式の除去を適用することが効果的である。共通部分式の除去によって取り除くことができる計算は、すべての実行経路上において冗長なものに限られる。一方、部分冗長除去法は、いくつかの実行経路で冗長であるが、他の実行経路では冗長でない部分冗長性を扱うことができる。部分冗長除去は、コード移動に基づくコード最適化なので、同時に命令スケジューリングが行える可能性が指摘されてきた。本研究では、部分冗長除去のアルゴリズムを冗長性の除去だけでなく、スケジューリング元として有効な命令をプログラム全体から選択し、適切な空きリソースへスケジュールする手法を提案する。本手法では、プログラム中のすべての空きリソースに命令が存在すると仮定して、部分冗長除去のデータフロー解析を適用する。実際のスケジューリングは、データフロー方程式の解として得られた除去可能な命令と挿入点を、それぞれスケジューリング元と先と見なして、プログラムを変形することによって実現する。実際に実装して評価を行い、有効性を示す。

## Practical Global Instruction Scheduling Based on Partial Redundancy Elimination

RYOTARO OIKAWA<sup>1,a)</sup> MUNEHIRO TAKIMOTO<sup>1</sup>

Presented: March 16, 2012

Instruction scheduling is one of the most effective techniques to increase instruction-level parallelism. Especially, the instruction scheduling which allows speculative execution is known as a technique which exposes more parallelism in a program. However, since the speculative execution can introduce new redundant operations as a second order effect, it is effective for the redundant operations to be eliminated using another code optimization technique such as common sub-expression elimination (CSE). The CSE eliminates the totally redundant operations which are redundant on all execution paths, but it cannot eliminate the partially redundant operations that are redundant on some paths and not on the other paths. On the other hand, inserting the same operations on the execution paths with no redundancy, the original operations become totally redundant, and therefore can be eliminated. Such a technique is called partial redundancy elimination (PRE). Since a combination of elimination and insertion corresponds to code motion, it has been suggested that the instruction scheduling can be achieved through PRE. We propose an approach that uses PRE algorithm not only to eliminate redundant operations but also to discover operations to be scheduled in entire program and schedule them to idle resources. Our approach is achieved by solving dataflow equations of PRE on the assumption that operations are putted on any idle resources, where the eliminated operations and the inserted points given as solutions respectively correspond to scheduled operations and idle resources to which be scheduled. We have implemented our approach as a code optimization phase, and conducted some experiments. We show the effectiveness of our approach through the experimental results.

<sup>1</sup> 東京理科大学  
Tokyo University of Science, Noda, Chiba 278-8510, Japan  
<sup>a)</sup> oikawa@cs.is.noda.tus.ac.jp